

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Využití nízko nákladového řešení pro vzdálenou správu
Using low-cost solution for remote management**

2019

Bc. Dalibor Fujeřík

Zadání diplomové práce

Student:

Bc. Dalibor Fujerík

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T059 Mobilní technologie

Téma:

Využití nízko nákladového řešení pro vzdálenou správu
Using low-cost solution for remote management

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem diplomové práce je navrhnout nízko nákladové řešení pro vzdálenou správu s využitím platformy Raspberry Pi. Obsah diplomové práce musí splňovat následující body zadání:

1. Studium možností vzdálené správy s využitím nízko nákladového řešení na bázi Raspberry Pi.
2. Instalace modemu a konfigurace skriptů na Raspberry Pi pro automatické připojení k serveru s využitím OpenVPN.
3. Popis instalace serverové platformy s OpenVPN a s databází připojených klientů.
4. Popis instalace kamery na Raspberry Pi s možností vzdáleného sledování obrazu.
5. Zpracované možnosti emulace PS2 nebo USB klávesnice, popř. myši na Raspberry Pi.
6. Analýza možnosti vzdálené konfigurace síťových zařízení skrze RJ-45 konektor na Raspberry Pi.
7. Příprava podrobné dokumentace navrženého řešení.

Seznam doporučené odborné literatury:

[1] Steve McCarthy: Raspberry Pi 3: Setup, Programming and Developing Amazing Projects with Raspberry Pi for Beginners - With Source Code and Step by Step Guides, CreateSpace Independent Publishing Platform (May 12, 2017), ISBN-13: 978-1546685616.

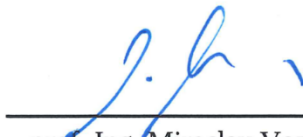
Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Lukáš Kapičák**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



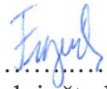

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlásenie študenta

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave dňa: 29. apríla 2019


.....
podpis študenta

Pod'akovanie

Rád by som pod'akoval Ing. Lukášovi Kapičákovi za odbornú pomoc a konzultáciu pri vytváraní tejto diplomovej práce.

Abstrakt

Táto diplomová práca obsahuje návrh nízko nákladového riešenia pre vzdialenú správu. Vzniknuté zariadenie využíva kombináciu Raspberry Pi a Arduina. Obsahuje implementáciu Webového rozhrania, dvoch programov pre Raspberry Pi a Arduino. Pomocou Webového rozhrania je možné celé zariadenie ovládať z pohodlia domova, bez ohľadu na to, kde sa fyzicky zariadenie nachádza. Webové rozhranie bolo napísané JavaScriptovým frameworkom AngularJS. Na Raspberry Pi je umiestnená kamera, pomocou ktorej je umožnené vzdialené sledovanie displeja pripojeného počítača. Aplikácia pre Raspberry Pi napísaná v jazyku Java, sa stará o celú logiku komunikácie s Webovým rozhraním a Arduino. Komunikácia s Webovým rozhraním je zabezpečená WebSocket protokolom a komunikácia s Arduino I2C zbernicou. Druhý program pre Arduino je napísané v jazyku C a zabezpečuje emuláciu klávesnice a myši na cieľovom zariadení. Pre vzniknuté zariadenie bola vytvorená krabička pomocou 3D tlačiarne.

Kľúčové slová

Raspberry Pi; Arduino; OpenVPN; GPIO; Webové rozhranie; emulácia klávesnice; emulácia myši;

Abstract

This thesis contains a proposal for a low cost remote management solution. The device uses a combination of Raspberry Pi and Arduino. It includes Web interface implementation, two Raspberry Pi and Arduino programs. By using the Web Interface can be the entire device controlled from home comfort, no matter where the device is located. The web interface was written by the JavaScript framework AngularJS. A Raspberry Pi camera is installed to allow remote viewing of the connected computer. The Raspberry Pi application written in Java takes care of the whole communication logic with Web interface and Arduino. Web interface communication is secured by WebSocket protocol and communication with Arduino I2C bus. The second Arduino program is written in language C and provides keyboard and mouse emulation on the target device. For the device was created a box by 3D printer.

Key words

Raspberry Pi; Arduino; OpenVPN; GPIO; Web interface; Keyboard emulation; Mouse emulation;

Zoznam použitých skratiek

Skratka	Význam
IT	Information technology
KVM	Keyboard, video and mouse
UTP	Unshielded twisted pair
USB	Universal Serial Bus
BIOS	Basic Input/Output System
TCP	Transmission Control Protocol
IP	Internet Protocol
PIN	Personal identification number
RFB	Remote framebuffer
VPN	Virtual private network
LTE	Long-Term Evolution
HDMI	High-Definition Multimedia Interface
GPIO	General-purpose input/output
RAM	Random Access Memory
4K	4K resolution
DSI	Display Serial Interface
CSI	Camera Serial Interface
IoT	Internet of things
HD	High-definition video
LCD	Liquid-crystal display
PWM	Pulse width modulation
EEPROM	Electrically erasable programmable read-only memory
NAS	Network access server
LED	Light-emitting diode
UART	Universal asynchronous receiver-transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit

WPA2	Wi-Fi Protected Access II
WEP	Wired Equivalent Privacy
HID	Human interface device
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
SPA	Single-page application
GUI	Graphical user interface
UI	User interface
GND	Ground
TLS	Transport Layer <i>Security</i>
SSL	Secure Sockets Layer
SSID	Service Set Identifier
EDGE	Enhanced Data rates for GSM Evolution
CDMA	Code-division multiple access
APN	Access Point Name
NAT	Network address translation
VPS	Virtual private server
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
GIT	Distributed version-control
VCS	Version control
OS	Operating system
HTTP	Hypertext Transfer Protocol
IDE	Integrated development environment
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol

Obsah

Úvod.....	- 13 -
1 Vzdialená správa	- 14 -
1.1 Čo je to vzdialená správa	- 14 -
1.2 Existujúce riešenia.....	- 14 -
1.2.1 Hardvérové riešenia.....	- 14 -
1.2.2 Softvérové riešenie	- 16 -
1.3 Vlastné riešenie	- 16 -
2 Výber vhodného hardvéru.....	- 17 -
2.1 Predpoklady hardvérových nárokov.....	- 17 -
2.2 Dostupné nízko nákladové mikropočítače	- 17 -
2.2.1 UP2 Squared.....	- 18 -
2.2.2 Huawei HiKey 960.....	- 18 -
2.2.3 ODROID-XU4	- 19 -
2.2.4 BeagleBone Black	- 20 -
2.2.5 Asus Tinker Board.....	- 21 -
2.2.6 Imagination Creator Ci20.....	- 21 -
2.2.7 Raspberry Pi 3	- 22 -
2.3 Výber vhodného zariadenia.....	- 23 -
2.4 Raspberry Pi 3	- 23 -
2.4.1 Operačný systém pre Raspberry Pi.....	- 24 -
2.4.2 Možnosti využitia Raspberry Pi	- 24 -
2.4.3 Dotykový displej a kamera.....	- 25 -
2.4.4 GPIO piny.....	- 25 -
2.4.5 Napájanie.....	- 26 -
2.5 Výber napájania.....	- 27 -
2.6 Výber kamery	- 27 -
2.7 USB modem	- 28 -
2.8 Hardvér pre emuláciu klávesnice a myši.....	- 28 -
2.9 Arduino	- 29 -

2.9.1	Bootloader	- 29 -
2.9.2	Arduino Shields.....	- 29 -
2.9.3	Arduino Leonardo	- 30 -
3	Návrh riešenia	- 31 -
3.1	Analýza požadovanej funkcionality	- 31 -
3.2	Princíp komunikácie.....	- 31 -
3.3	Webová aplikácia	- 33 -
3.3.1	Použitý framework	- 33 -
3.3.2	AngularJS	- 33 -
3.3.3	Grafické užívateľské rozhranie (GUI).....	- 34 -
3.3.4	Navigačná lišta	- 34 -
3.3.5	Prihlasovacia obrazovka	- 34 -
3.3.6	Zoznam pripojených klientov.....	- 34 -
3.3.7	Interface ovládanie klienta	- 35 -
3.4	Aplikácia pre Raspberry Pi.....	- 35 -
3.4.1	I2C zbernica	- 35 -
3.4.2	WebSocket.....	- 36 -
3.4.3	Program Motion	- 37 -
3.5	Program na emuláciu klávesnice a myši	- 37 -
3.6	Krabička pre zariadenie.....	- 37 -
3.7	Zautomatizovanie Raspberry Pi	- 37 -
4	Konfigurácia USB modemu	- 38 -
4.1	Použitý modem.....	- 38 -
4.2	Konfigurácia.....	- 38 -
5	Inštalácia serverovej platformy s OpenVPN	- 40 -
5.1	Operačný systém servera.....	- 40 -
5.2	OpenVPN	- 40 -
5.2.1	Čo je to OpenVPN.....	- 40 -
5.2.2	Inštalácia a konfigurácia OpenVPN	- 41 -
6	Inštalácia kamery s umožnením streamovania	- 44 -
6.1	Inštalácia kamery.....	- 44 -

6.2	Program Motion - vzdialené sledovanie obrazu	45 -
7	Implementácia webového rozhrania pre serverovú platformu	46 -
7.1	Vývojové prostredie WebStorm	46 -
7.2	Samotná webová aplikácia	47 -
7.2.1	Prihlásenie	47 -
7.2.2	Získanie zoznamu pripojených klientov	48 -
7.2.3	Ovládanie klienta	49 -
7.3	Nasadenie na VPS Server	51 -
7.3.1	Čo je to Apache2 Web Server	51 -
7.3.2	Konfigurácia Apache2 Web Serveru	51 -
8	Implementácia Raspberry Pi aplikácie	52 -
8.1	Vývojové prostredie IntelliJ IDEA	52 -
8.2	Samostatná aplikácia	53 -
8.2.1	Java knižnica pre I2C zbernicu	53 -
8.2.2	Komunikácia medzi Webovým rozhraním a Raspberry Pi	54 -
8.2.3	Komunikácia medzi Raspberry Pi a Arduino Leonardo	55 -
8.3	Automatizácia Raspberry Pi	56 -
8.3.1	Automatické pripojenie ku Internetu	56 -
8.3.2	Automatické spustenie Aplikácie pre komunikáciu	57 -
8.3.3	Automatické pripojenie na OpenVPN server	57 -
9	Implementácia programu na emuláciu klávesnice a myši cez Arduino Leonardo	58 -
9.1	Vývojové prostredie Arduino IDE	58 -
9.2	Použité knižnice	58 -
9.2.1	Keyboard	59 -
9.2.2	Mouse	60 -
9.3	Program pre emuláciu klávesnice a myši	60 -
10	Analýza možnosti konfigurácie sieťových zariadení	61 -
10.1	Štandardné sieťové prvky	61 -
10.2	Sieťové nástroje	61 -
10.3	Možnosti konfigurácie	61 -
11	Dosiahnuté výsledky	62 -

11.1	Praktická použiteľnosť	- 62 -
12	Záver	- 64 -
12.1	Možné rozšírenia výsledného riešenia	- 64 -
	Použitá literatúra	- 65 -
	Zoznam príloh	- 67 -

Úvod

Cieľom tejto práce bolo vytvoriť riešenie podľa zadania. Hlavnou úlohou vytvoreného riešenia je pomocou Raspberry Pi a Arduina umožniť vzdialenú správu. Výsledné riešenie bude možné ovládať pomocou Webového rozhrania. Webové rozhranie bude zobrazovať databázu pripojených zariadení k OpenVPN serveru. Používateľ sa bude môcť pripojiť na zariadenie, vidieť obraz z kamery a ovládať klávesnicu a myš na cieľovom zariadení.

V kapitole 1 sa zaoberám problematikou vzdialenej správy. Prinášam pohľad na existujúce hardvérové a softvérové riešenia pre vzdialenú správu a pohľad na základný náčrt výsledného riešenia. V kapitole 2 sa zaoberám výberom správneho mikropočítača a jednotlivých periférií pre Raspberry Pi. V kapitole 3 sa venujem návrhu výsledného riešenia, ujasnením požiadavkov a princípu jeho fungovania. Kapitola 4 sa zaoberá konfiguráciou USB modemu pre využitie mobilného internetu. V kapitole 5 sa venuje konfigurácii serverovej platformy a konfigurácii OpenVPN serveru. Kapitola 6 sa zaoberá popisom inštalácie kamery na Raspberry Pi s možnosťou vzdialeného sledovania obrazu. Kapitola 7 sa venuje implementácií webového rozhrania, ktoré obsahuje databázu pripojených klientov ku OpenVPN serveru a umožňuje následne cez Webové rozhranie pripojeného klienta vzdialene ovládať. V kapitole 8 sa venujem implementácií aplikácie pre Raspberry Pi, ktorá zabezpečuje komunikáciu medzi Webovým rozhraním a Arduinom. Kapitola 9 rieši implementáciu programu pre Arduino, ktorý zabezpečuje emuláciu klávesnice a myši. Kapitole 10 analyzuje možnosti vzdialeného konfigurovania sieťových zariadení. V kapitole 11 je súhrn dosiahnutých výsledkov a záver je v kapitole 12.

1 Vzdialená správa

Táto kapitola prináša pohľad na problematiku vzdialenej správy, prehľad existujúcich riešení slúžiacich na vzdialenú správu, ich výhody a nevýhody. Taktiež obsahuje výhody a nevýhody môjho riešenia a základný náčrt jeho fungovania.

1.1 Čo je to vzdialená správa

Vzdialená správa sa dá definovať ako schopnosť programu alebo zariadenia ovládať iné zariadenie zo vzdialeného miesta. Umožňuje vykonávať prácu a pristupovať k údajom, ktoré sú uložené na domácom zariadení, keď cestujeme.

Jedným z hlavných cieľov vzdialeného ovládania počítača je eliminácia potreby akéhokoľvek fyzického zásahu. Po vytvorení vzdialeného pripojenia medzi lokálnym a vzdialeným počítačom môžeme plnohodnotne ovládať a pristupovať k vzdialenému zariadeniu akoby sme sedeli fyzicky pred ním. Môžeme spravovať a používať aplikácie, odstraňovať problémy a dokonca vykonávať administratívne úlohy.

Stále viac priemyselných odvetí začína využívať vzdialené ovládanie počítača na zlepšenie svojich pracovných funkcií. Moderné podniky využívajú túto technológiu na maximalizáciu efektívnosti a zvýšenie produktivity. V oblasti IT¹ je táto technológia využívaná na poskytovanie podpory pre klientov mimo pracoviska. Vďaka vzdialenej správe nie je nutné vyslať technika za klientom. Môže jednoducho pracovať na akýchkoľvek technických otázkach bez nutnosti opustenia svojho skutočného pracovného miesta. Nakoniec to ušetrí veľa peňazí, času a úsilia.

1.2 Existujúce riešenia

Na trhu môžeme nájsť niekoľko riešení vzdialenej správy počítača. Táto podkapitola sa venuje existujúcim hardvérovým a softvérovým riešeniam.

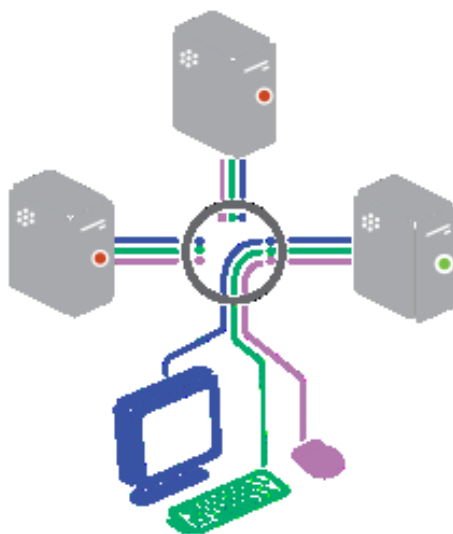
1.2.1 Hardvérové riešenia

KVM Switch je hardvérové riešenie. Jeho skratka KVM vychádza zo slov: klávesnica, video a myš. Umožňuje užívateľovi ovládať viacero počítačov z jedného monitoru, klávesnice a myši. Umožňujú fungovať aj v opačnom smere a teda pripojenie viacerých klávesníc, myší a monitorov ku jednému počítaču a spravovať teda jeden počítač z viacerých miest. Viac sa možno dočítať na [1].

¹ Informačné Technológie

KVM Switch-e existujú v dvoch variantoch:

- **Lokálne KVM:** Umožňuje ovládanie počítača až do vzdialenosti 300 metrov od užívateľskej konzoly (monitor, klávesnica a myš). Funguje skrz UTP² kábel kategórie 5. KVM zariadenia využívajúce USB sú schopné ovládať klávesnicu, monitor a myš až na vzdialenosť 5 metrov. Ovládanie prebieha v reálnom čase a nedochádza ku žiadnemu pozorovateľnému oneskoreniu pri komunikácii s cieľovým počítačom.
- **KVM cez IP:** Využívajú počítačovú sieť. Pomocou špeciálnych mikrokontrolérov a špecializovaného hardvéru zachytávajú signály videa, klávesnice a myši. Potom tieto signály komprimujú a prevádzajú na IP pakety, ktoré sú poslané cez počítačovú sieť na vzdialenú konzolu s aplikáciou, ktorá ich spracováva a zostaví dynamický obraz. KVM je väčšinou pripojené k systému neustále, takže je k dispozícii v priebehu štartu počítača alebo ku ovládaniu BIOS-u³. Umožňuje ovládať viacero počítačov cez rozsiahle počítačové siete, lokálne siete alebo telefónne linky pomocou protokolu TCP/IP. Tieto KVM sú obmedzené priepustnosťou siete, takže môže dochádzať ku oneskoreniam pri komunikácii s cieľovým počítačom.



Obrázok 1.1: Jednoduché znázornenie KVM switch-a.

Zdroj: Wikipedia, rok 2019

Veľkou nevýhodou tohto riešenia je jeho cena, pre bežného užívateľa je príliš vysoká. Toto riešenie je vhodné skôr pre veľké firmy. Taktiež za nevýhodu považujem nutnosť vybudovania sieťovej infraštruktúry a inštaláciu nutného hardvéru do siete pre ich fungovanie.

² Netienená krútená dvojlinka

³ Základný vstupno-výstupný systém počítača

1.2.2 Softvérové riešenie

Softvérových riešení dnes (v roku 2019) existuje niekoľko. Väčšina je pre domáce použitie zadarmo. Sú užívateľsky priateľské, nainštalovať a nastaviť ich dokáže úplný nováčik a zaberie to len niekoľko kliknutí.

Niekoľko softvérových riešení:

- **TeamViewer:** Veľmi obľúbený. Možné jednorazové použitie bez nutnosti registrácie a inštalácie. Je multiplatformový a má jednoduché ovládanie. Umožňuje prenos súborov. Pri vzdialenej pomoci umožňuje chat v textovej podobe alebo pomocou kamery a mikrofónu. Pre nekomerčné použitie je k dispozícii zadarmo. Viac na [2].
- **Vzdialená plocha (Windows):** Tento nástroj je priamo súčasťou Windows-u a nie je nutné inštalovať ďalší software⁴. Umožňuje sa pripojiť cez IP adresu na vzdialenú plochu. Ďalej je možnosť aj vzdialenej pomoci. Tento úkon zvládne aj neskúsený užívateľ vďaka prehľadnému pomocníkovi. Viac na [3].
- **Vzdialená plocha Chrome:** Užívatelia prehliadača Chrome si môžu nainštalovať software pre vzdialenú plochu priamo z obchodu Chrome. Pripájanie funguje pomocou PIN kódu. Pripojenie je možné z ľubovoľného počítača s Chrome a konfigurácia je taktiež veľmi jednoduchá. Okrem počítača je dostupná aj pre tablety a smartfóny. Viac na [4].
- **RealVNC:** Funguje na protokole RFB a je možné ho použiť na všetkých platformách. Pre osobné použitie je zadarmo. To však neponúka šifrovaný prenos a preto sa odporúča použiť ho len v lokálnej sieti alebo v kombinácii s VPN. Viac na [5].

Hlavnou nevýhodou všetkých týchto softvérových riešení je nutnosť fungujúceho operačného systému, na ktorom je nutné softvér spustiť. Vzdialená správa funguje až po nabehnutí operačného systému a nie je teda možné riešiť problémy vyskytujúce sa pri štarte počítača.

1.3 Vlastné riešenie

Výsledné riešenie diplomovej práce odstraňuje nevýhody existujúcich riešení. Pri vzdialenej správe pomocou výsledného zariadenia, nie je nutné mať funkčný operačný systém. Taktiež cena nebude príliš vysoká vďaka použitiu nízko nákladového mikropočítača. Nie je nutné ani upravovať sieť u potencionálneho zákazníka. Zariadenie je úplne samostatné a stačí ho pripojiť iba ku vzdialenému počítaču. Ovládanie prebieha pohodlne cez webové rozhranie, nie je nutné nič viac inštalovať do počítača.

⁴ Počítačový program

2 Výber vhodného hardvéru

Táto kapitola sa zaoberá výberom správneho hardvéru pri zohľadnení požadovaných nárokov. Prináša pohľad na rôzne mikropočítače na báze Raspberry Pi. Ďalej sa zaoberá výberom kamery, modemu a hardvéru nutnému ku emulácii klávesnice a myši.

2.1 Predpoklady hardvérových nárokov

Predpokladané nároky na hardvér:

- **Výkon:** Požiadavky na výkon zariadenia nie sú príliš vysoké. Neočakávajú sa žiadne náročné výpočtové operácie. Postačuje splnenie minimálnych požiadavkov na inštaláciu jednej z distribúcií Linuxu a možnosť spustenia viacerých systémových procesov súčasne.
- **Internetová konektivita:** Pre čo najväčšie pokrytie univerzálnosti v konektivitě zariadenia, je nutné aby hardvér disponoval s Ethernetovým konektor, ďalej s modemom Wi-Fi a USB portom pre možnosť pripojenie LTE modemu.
- **Porty:**
 - Minimálne dva USB porty pre možnosť pripojenia modemu alebo prípadnej kamery cez USB.
 - HDMI port, ktorý je vhodný na prvotnú konfiguráciu, avšak nie je nutnosťou.
 - RJ45 konektor pre Ethernetové pripojenie a umožnenie konfigurácie sieťových zariadení.
 - GPIO porty, ktoré sú predpokladom emulácie klávesnice a myši, prípadne pripojenia ďalšej dosky, ktorej potreba môže vzniknúť na základe vývoja daného zariadenia.
 - Kamerový port pre pripojenie kamery cez špecializovaný port a nevyužívať tak USB zbernicu.
 - Port pre SD kartu na umožnenie inštalácie operačného systému.
- **Cena:** Dôležitým aspektom výberu hardvéru bude čo najnižšia možná cena.

2.2 Dostupné nízko nákladové mikropočítače

Dnes (v roku 2019) môžeme na trhu nájsť niekoľko alternatív nízko nákladových mikropočítačov. Každý z nich ponúka vlastné výhody a sú vhodné pre rôzne riešenia. Táto kapitola sa venuje porovnaniu mikropočítačov dostupných na trhu, zohľadnením ich výhod a nevýhod. Taktiež obsahuje prieskum trhu potencionálne vhodnejšieho mikropočítača než je Raspberry Pi.

V nasledujúcich podkapitolách sú spomenuté niektoré z najlepších alternatív nízko nákladových mikropočítačov. Zamerané sú na základný popis jednotlivých mikropočítačov a ich špecifikáciu.

2.2.1 UP2 Squared



Obrázok 2.1: *UP2 Squared*

Zdroj: AAEON, rok 2019

UP2 Squared je mikropočítač, ktorý by sa dal zaradiť medzi hobby dosky a skutočné mikropočítače. V porovnaní s Raspberry Pi má obrovský výkon, predáva sa v rôznych špecifikáciách až po 4-jadrový procesor Intel Pentium N4200 s frekvenciou 2.5 GHz a 8 GB pamäte RAM. Výkonom sa teda blíži k cenovo dostupnejším notebookom. Na tejto špecifikácii sa odráža aj cena, ktorá sa vyšplhá na približne 330€. Hlavnou výhodou oproti konkurencii je to, že vďaka procesoru Intel podporuje inštaláciu plnohodnotného Windows 10. Viac na [6].

Tabuľka 2.1: *Špecifikácie UP2 Squared*

Procesor	Intel Pentium N4200 2.5 GHz
RAM	8 GB LPDDR4
Porty	3x USB 3.0, 1x HDMI, 1x Displej Port, 1x SATA
Konektivita	2x Ethernet
Rozmery	86 x 90 mm
Cena	330€

Zdroj: Vlastné spracovanie autora.

2.2.2 Huawei HiKey 960



Obrázok 2.2: *HiKey 960*

Zdroj: ZDNet, rok 2019

Mikropočítač HiKey960 od Huawei-u je prémiová doska, na ktorej môže bežať systém Linux, podporuje však aj možnosti inštalácie Androidu. Hardvérové vybavenie dosky sa rovná dnešným smartfónom. Procesorom je tu Kirin 960, pamäť RAM je 3 GB a disponuje s grafickým čipom. Taktiež obsahuje 40-pinový a 60-pinový konektor na pripojenie hardvéru, ako je napríklad kamera. Viac na [7].

Tabuľka 2.2: Špecifikácia Huawei HiKey 960

Procesor	Huawei Kirin 960 2.36 GHz
RAM	3 GB LPDDR4 SDRAM
Porty	2x USB 3.0, 1x USB-C 2.0, 1x HDMI
Konektivita	Dual-band Wi-Fi, Bluetooth 4.1
Rozmery	85 x 55 mm
Cena	230€

Zdroj: Vlastné spracovanie autora.

2.2.3 ODDROID-XU4



Obrázok 2.3: Odroid-XU4

Zdroj: Génération Robots, rok 2019

Odroid-XU4 disponuje pôsobivým výkonom, ktorý zabezpečuje osem jadrový procesor od Samsungu, ktorému sekunduje 2 GB RAM. Toto zariadenie sa výkonom približuje ku stolným počítačom. Disponuje gigabitovým Ethernetovým pripojením, ale chýba mu bezdrôtová konektivita. Na doske je možné spustiť rôzne verzie Linuxu, ako je Ubuntu 16.04. Taktiež je podporovaný aj systém Android. Viac na [8].

Tabuľka 2.3: Špecifikácia Odroid-XU4

Procesor	Samsung Exynos 5422 2.1 GHz
RAM	2 GB LPDDR3
Porty	2x USB 3.0, 1x USB 2.0, 1x HDMI
Konektivita	Ethernet
Rozmery	83 x 58 mm
Cena	50€

Zdroj: Vlastné spracovanie autora.

2.2.4 BeagleBone Black



Obrázok 2.4: BeagleBone Black

Zdroj: BeagleBoard.org, rok 2019

BeagleBone Black patrí medzi cenovo dostupnejšie dosky. Svojimi špecifikáciami je podobná špecifikáciám Raspberry Pi. Použitý ARM procesor má frekvenciu 1 GHz. Obsahuje digitálne a analógové vstupno-výstupné piny a Ethernetový konektor.

Tabuľka 2.4: Špecifikácia BeagleBone Black

Procesor	AM335x ARM Cortex-A8 1 GHz
RAM	512 MB DDR3
Porty	1x USB 2.0, 1x HDMI
Konektivita	Ethernet
Rozmery	86 x 53 mm
Cena	45€

Zdroj: Vlastné spracovanie autora.

2.2.5 Asus Tinker Board



Obrázok 2.5: *Asus Tinker Board*

Zdroj: Curry's, rok 2019

Asus Tinker je označovaný ako klon Raspberry Pi, vyzerá skoro identicky ako Raspberry Pi, až na rôzne umiestnenie portov. Asus Tinker je však výkonnejší ako Raspberry Pi a podporuje rozlíšenie až v 4K. Cena je skoro dvojnásobná ako cena Raspberry Pi. Viac na [10].

Tabuľka 2.5: *Špecifikácia Asus Tinker Board*

Procesor	Rockchip RK3288 1.8 GHz
RAM	2 GB DDR3
Porty	4x USB 2.0, 1x HDMI
Konektivita	802.11n Wi-Fi, Bluetooth 4.0, Ethernet
Rozmery	85 x 54 mm
Cena	70€

Zdroj: Vlastné spracovanie autora.

2.2.6 Imagination Creator Ci20



Obrázok 2.6: *Imagination Creator Ci20*

Zdroj: eLinux.org, rok 2019

Imagination Creator Ci20 je doska vhodná na aplikáciu v IoT. Disponuje 8GB Flash pamäťou, na ktorej sa nachádza predinštalovaný Debian Linux. Podporuje taktiež aj systém Android. Z internetovej konektivity disponuje Ethernetovým konektorom, Wi-Fi a Bluetooth. Procesor má frekvenciu 1.2 GHz a pamäť RAM 1 GB. Viac na [11].

Tabuľka 2.6: Špecifikácia Imagination Creator Ci20

Procesor	Ingenic JZ4780 1.2 GHz
RAM	1 GB DDR3 SDRAM
Porty	2x USB, 1x HDMI
Konektivita	802.11n Wi-Fi, Bluetooth 4.0, Ethernet
Rozmery	100 x 92 mm
Cena	70€

Zdroj: Vlastné spracovanie autora.

2.2.7 Raspberry Pi 3



Obrázok 2.7: Raspberry Pi 3

Zdroj: Amazon.in, rok 2019

Raspberry Pi 3 je nový model z rodiny Raspberry Pi dosiek. Je to malý mikropočítač s veľkosťou kreditnej karty. Môže slúžiť ako stolový počítač alebo riadiť chytrú domácnosť. Disponuje bezdrôtovou konektivitou, má štvorjadrový procesor s frekvenciou 1.4 GHz a pamäť RAM 1GB. Jeho výhodou je kompatibilita s príslušenstvom (rozmerovo a rozložením konektorov), s predchádzajúcimi verziami Raspberry Pi. Podporuje rôzne distribúcie Linuxu. Ďalej obsahuje audio výstup a GPIO konektor.

Tabuľka 2.7: Špecifikácia Raspberry Pi 3

Procesor	Broadcom 1.4 GHz
RAM	1 GB DDR3
Porty	4x USB 2.0, 1x HDMI
Konektivita	802.11ac Wi-Fi, Bluetooth 4.2, Ethernet
Rozmery	85 x 56 mm
Cena	40€

Zdroj: Vlastné spracovanie autora.

2.3 Výber vhodného zariadenia

Všetky spomínané mikropočítače sú porovnané v nasledujúcej tabuľke.

Tabuľka 2.8: Porovnávací tabuľka mikropočítačov

	UP2 Squared	Huawei HiKey 960	Odroid- XU4	BeagleBone Black	Asus Tinker Board	Imagination Creator Ci20	Raspberry Pi 3
Minimálne požiadavky pre Linux	Áno	Áno	Áno	Áno	Áno	Áno	Áno
Ethernet	Áno	Nie	Áno	Áno	Áno	Áno	Áno
Wi-Fi	Nie	Áno	Nie	Nie	Áno	Áno	Áno
Bluetooth	Nie	Áno	Nie	Nie	Áno	Áno	Áno
USB porty	Áno 3x	Áno 2x	Áno 3x	Áno 1x	Áno 4x	Áno 2x	Áno 4x
HDMI	Áno	Áno	Áno	Áno	Áno	Áno	Áno
Kamera port	Áno	Áno	Nie	Nie	Áno	Áno	Áno
GPIO port	Áno	Áno	Áno	Áno	Áno	Áno	Áno
MicroSD	Iné	Áno	Áno	Áno	Áno	Áno	Áno
Cena	330€	230€	50€	45€	70€	70€	40€

Zdroj: Vlastné spracovanie autora.

Z porovnávací tabuľky môžeme vidieť, že najviac vyhovujú predpokladaným požiadavkám tri zariadenia a to Asus Tinker Board, Imagination Creator Ci20 a Raspberry Pi 3. Raspberry Pi 3 vychádza cenovo ako najdostupnejšie riešenie, ktoré spĺňa všetky predpokladané požiadavky. Môžeme teda konštatovať, že použitie Raspberry Pi 3 je v našom riešení najlepšou cestou a v súčasnosti (v roku 2019) sa nenachádza na trhu nič vhodnejšie.

2.4 Raspberry Pi 3

Raspberry Pi je malý mikropočítač o veľkosti kreditnej karty. Vytvorila ho britská nadácia Raspberry Pi Foundations s cieľom podporiť výučbu programovania a informatiky na školách. Viac sa možno dočítať vo [27].



Obrázok 2.8: Raspberry Pi 3 model B + pohľad z vrchu

Zdroj: BuyaPi.ca, rok 2019

Na trhu je k dispozícii niekoľko verzií Raspberry Pi, ktoré sa líšia od seba veľkosťou, výkonom, počtom GPIO pinov, počtom USB portov, prítomnosťou Ethernetového konektoru, Wi-Fi alebo Bluetooth modulu.

Raspberry Pi 3 model B, ktorý bol vybraný pre riešenie diplomovej práce. Disponuje konektorom HDMI rozhrania, audio výstupom a špeciálnym GPIO konektorom, ktorý slúži pre pripojenie rôznych periférií. Ďalej obsahuje 4x USB konektor, Ethernet, Wi-Fi, Bluetooth, Camera Serial Interface (CSI) pre možnosť pripojenia kamery a Display Serial Interface (DSI) pre pripojenie displeja. Zariadenie disponuje microSD slotom. O výkon sa postará procesor Broadcom Cortex A53, ktorý má frekvenciu 1.2GHz a 1GB operačnej pamäte, ktorá na bežné používanie postačuje. O grafické operácie sa stará integrovaná grafická karta Broadcom VideoCore IV. Viac v [12].

Ďalším krokom je výber operačného systému pre Raspberry Pi, možnosti využitia Raspberry Pi, možnosti pripojenia kamery, detailný popis GPIO pinov, pripojenie periférií a popis možností napájania.

2.4.1 Operačný systém pre Raspberry Pi

Operačný systém Raspberry Pi sa inštaluje na pamäťovú kartu. Staršie modely používali typ klasickej SD karty, novšie už používajú typ microSD. Na inštaláciu systému budeme potrebovať iný počítač s čítačkou kariet SD.

Raspberry Pi by mal pracovať s akoukoľvek kompatibilnou kartou SD, aj keď existujú určité odporúčania, ktoré by sa mali dodržiavať. Pre inštaláciu NOOBS alebo inštaláciu obrazu Raspbian je minimálna odporúčaná veľkosť karty 8 GB. Pre inštaláciu Raspbian Lite je odporúčaná veľkosť minimálne 4 GB. Niektoré distribúcie, však dokážu bežať na oveľa menších kartách. Ďalším odporúčaním je vybrať správnu Class karty. Class karty určuje rýchlosť zápisu na kartu. Karta Class 4 bude schopná zapisovať 4 MB / s, zatiaľ čo Class 10 by mala byť schopná dosiahnuť 10 MB / s. Treba však poznamenať, že to neznamena, že karta Class 10 bude všeobecne lepšia na použitie ako karta Class 4, pretože táto rýchlosť zápisu sa často dosahuje za cenu nižšej rýchlosti čítania a zvýšenej doby vyhľadávania.

Väčšina dostupných operačných systémov pre Raspberry Pi je založená na Linuxe. Oficiálnym podporovaným operačným systémom je Raspbian, ktorý je založený na verzii Linuxu Debian. Raspbian obsahuje množstvo predinštalovaného softvéru pre vzdelávanie, programovanie a všeobecné použitie. Má Python, Scratch, Sonic Pi, Java a ďalšie. Ďalším zaujímavým operačným systémom je Ubuntu Mate, ktorý taktiež prichádza s množstvom predinštalovaného softvéru. Taktiež je možné na Raspberry Pi nainštalovať Windows 10 IoT, ktorý je založený na Windows 10 a je určený pre IoT.

2.4.2 Možnosti využitia Raspberry Pi

Raspberry Pi bolo vytvorené s cieľom podporiť výučbu programovania a informatiky na školách. Či už na internete alebo v knižnej podobe existuje veľa návodov pre využitie a vytváranie projektov s Raspberry Pi.

Jedným z populárnych riešení využitia Raspberry Pi je pripojenie cez HDMI ku televízii a využitie ako multimediálne centrum. So softvérom XBMC je potom možné streamovať hudbu alebo video z lokálneho alebo vzdialeného úložiska. Funguje i s niektorými internetovými službami, podporuje ovládanie diaľkovým ovládačom a Full HD video.

Ďalším zaujímavým riešením je využitie niekoľkých Raspberry Pi ako výkonný superpočítač. Ďalej je možné využitie Raspberry Pi ako NAS server a mať tak USB Flash disk alebo externý disk sprístupnený skrz sieť alebo pomocou Apache web server vytvoriť Web server.

Veľkou výhodou Raspberry Pi je, že vďaka pinom GPIO je možné pripojiť veľké množstvo elektronických súčiastok a senzorov. Následne sa dá využívať na meranie teploty v miestnosti. Veľmi obľúbeným využitím Raspberry Pi je pri dokúpení kamerového modulu, vytvorenie IP kamery a sledovať tak to, čo sa deje v dome počas užívateľovej neprítomnosti, a tak sa dajú napríklad sledovať domáci miláčikovia. Vďaka veľkému množstvu vstupno-výstupných pinov je možné Raspberry Pi zakomponovať do veľkého množstva robotických projektov. Jedna z pokročilých možností využitia, je ich využitie na automatizáciu domácností. Dá sa docieľiť ovládanie svetiel alebo spotrebičov v dome len pomocou mobilného telefónu s internetom. Dá sa nastaviť automatické vypínanie svetiel, ovládanie centrálneho kúrenia a veľa ďalších možností.

2.4.3 Dotykový displej a kamera

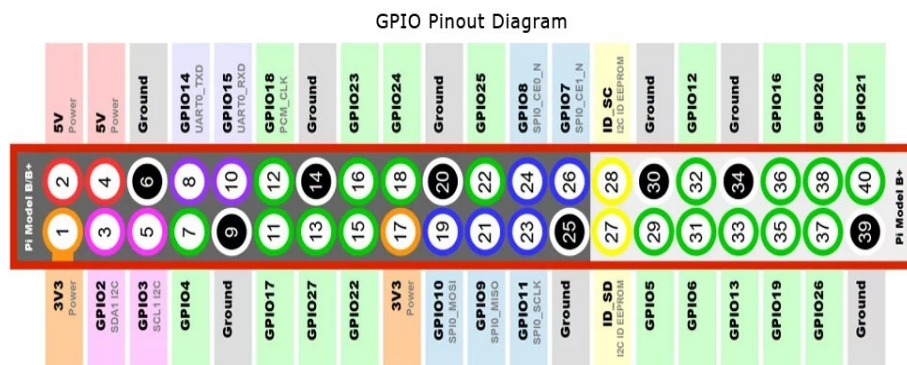
Dotykový displej pre Raspberry Pi sa pripája cez konektor DSI (Display Serial Interface), ktorý je umiestnený na doske. V niektorých prípadoch je tak možné používať dotykový displej pripojený cez DSI a LCD displej pripojený cez HDMI súčasne.

Modul Raspberry Pi Kamera je oficiálnym produktom nadácie Raspberry Pi Foundation. Pôvodný 5-megapixelový model bol vydaný v roku 2013 a v roku 2016 bol vydaný 8-megapixelový modul v2. Pre kameru sa využíva konektor CSI (Camera Serial Interface).

Pri vytváraní projektov sa odporúča použiť práve tieto konektory, namiesto HDMI alebo USB, kvôli obmedzenému počtu portov na doske.

2.4.4 GPIO piny

Najväčšou funkciou Raspberry Pi sú dva rady GPIO, univerzálnych vstupno-výstupných pinov nachádzajúcich sa v hornej časti dosky. Tento port má u Raspberry Pi 26 alebo 40 pinov. Periférie určené pre GPIO port, sú kompatibilné s oboma verziami, pretože 26 pinov je len kratšia verzia 40 pinovej verzie.



Obrázok 2.9: *Raspberry Pi GPIO piny*

Zdroj: Raspberry Pi Spy, rok 2019

Schéma rozmiestenia a poradia pinov je na obrázku 2.9. Na doske sa nachádzajú dva 5V a dva 3.3V piny, ďalej niekoľko uzemňovacích pinov, ktoré nie sú konfigurovateľné. Ostatné GPIO piny sú vstupno-výstupné a sú nastaviteľné na 3.3V alebo 0V. Vstup je tolerantný na 3.3V, pokiaľ by vstup prekročil 3.3V tak by mohlo dôjsť k poškodeniu dosky.

Niektoré piny však ponúkajú alternatívne funkcie, ktoré sú k dispozícii buď na všetkých pinoch alebo iba na špeciálnych pinoch.

- PWM(Impulzová šírková modulácia)
 - Softvérová PWM je dostupná na všetkých pinoch.
 - Hardvérová PWM je dostupná na pinoch GPIO12, GPIO13, GPIO18 a GPIO19.
- SPI zbernica
 - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1(GPIO7)
 - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C zbernica
 - Data: (GPIO2); Clock (GPIO3)
 - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Sériová linka
 - TX (GPIO14); RX (GPIO15)

2.4.5 Napájanie

Raspberry Pi 3 je napájané napätím 5.1V cez microUSB konektor. Presne koľko prúdu (mA) doska vyžaduje, závisí od toho, čo všetko je ku doske pripojené. Na oficiálnych stránkach je odporúčané použiť 2.5A napájanie, ktoré poskytne dostatok energie aj pri pripojení rôznych periférií ku doske.

Model B používa od 700 do 1000mA, taktiež v závislosti od toho, aké periférie sú pripojené ku zariadeniu. Model A dokáže pracovať len s 500mA bez pripojených periférií. Maximálny prúd môže byť 1A, ale ak by sme potrebovali pripojiť USB zariadenie, ktoré potrebuje viac prúdu ako je 1A, musíme použiť externý USB HUB so svojim napájaním.

Požiadavky na prúd pre Raspberry Pi sa zvyšujú od využívania rôznych rozhraní. Napríklad piny GPIO sú schopné bezpečne pracovať s 50mA rozdelenými do všetkých pinov. Jednotlivé GPIO piny dokážu pracovať iba s 16mA, HDMI port využíva 50mA, kamerový modul 250mA a klávesnica a myš môžu požadovať minimálne 100mA alebo viac ako 1A. Pre výber správneho napájania je teda nutné uvažovať, čo všetko bude ku doske pripojené a koľko prúdu jednotlivé časti potrebujú.

2.5 Výber napájania

Ako bolo spomínane v predchádzajúcej podkapitole v časti napájanie, pred výberom napájania je nutné uvažovať nad tým, čo všetko bude ku zariadeniu pripojené.

Prvou zvažovanou alternatívou bolo napájanie Raspberry Pi z Arduina, pomocou GPIO pinov a pinov na doske Arduina. Keďže Arduino bude pripojené k počítačom cez USB a teda bude napájané pomocou počítača. Týmto by sa využilo napájanie Arduina a nebolo by nutné mať ďalšie napájanie pre Raspberry Pi.

Napájanie však Raspberry Pi pomocou GPIO portov sa neodporúča, pretože ak by na piny prišlo viac ako 5.1V, došlo by k zničeniu Raspberry Pi. Ďalším obmedzením je, že Arduino napájané pomocou microUSB dokáže dať na výstup iba 500mA, čo je pre Raspberry Pi 3 Model B+ nedostačujúce. Preto bolo znevolené oficiálne napájanie pre Raspberry Pi.

2.6 Výber kamery

Keďže Raspberry Pi 3 disponuje s CSI (Camera Serial Interface), je najlepším riešením využiť práve tento port. Samostatný kamerový modul sa nachádza na malej doske a pripája sa pomocou krátkeho kábla. Kamera dokáže zachytávať video a robiť fotky. Kamera pripojená do CSI konektoru, oproti USB kamere, dokáže prenášať väčšie množstvo dát.

Modul Raspberry Pi Kamera je oficiálnym produktom od Raspberry Pi Foundation. Prvý model bol vydaný v roku 2013 a obsahoval 5 megapixelový snímač a v roku 2016 bol vydaný druhý model s 8 megapixelovým snímačom. Snímače použité v oboch modeloch sú porovnateľné s tými, ktoré sú použité v dnešných smartfónoch.

Tabuľka 2.9: *Hardvérová špecifikácia*

	Kamera Modul v1	Kamera Modul v2
Cena	25€	33€
Čip	OmniVision OV5647	Sony IMX219
Fotky	až 2592 x 1944 px	až 3280 x 2464 px
Video	1080p @ 30fps; 720p @ 60 fps; 640x480p @ 90 fps	1080p @ 30fps; 720p @ 60 fps; 640x480p @ 90 fps
Rozlíšenie	5 Megapixel	8 Megapixel

Zdroj: Vlastné spracovanie autora.

Ako výsledná kamera bude použitá Kamera Modul verzie 2, pretože využíva novší čip a cena oproti staršiemu modelu nie je až taká vysoká. Preto nie je dôvod nevyužiť a nezakúpiť nový model Kamerového modulu.

2.7 USB modem

Použitie USB modemu vo výslednom riešení, je nutné pre zabezpečenia internetovej konektivity, aj v prípade, ak nie je k dispozícii Wi-Fi alebo Ethernetové spojenie. Na modem nie sú špeciálne nároky a je možné použiť ktorýkoľvek dnes (v roku 2019) bežne predávaný LTE modem.

2.8 Hardvér pre emuláciu klávesnice a myši

Dôležitou vecou pri emulácii klávesnice a myši je pochopiť ako USB funguje. V počítačovej komunikácii funguje USB na modeli Master a Slave. To znamená že jedno zariadenie, zvyčajne osobný počítač, funguje ako Master a ovláda informačný tok z a do USB portu. Periférie ako je USB Flash disk alebo klávesnica fungujú ako Slave zariadenie. Slave zariadenia môžu byť obsluhované iba Master zariadením, pretože inak nebudú fungovať. Počítač ako Master zariadenie prijme Slave zariadenie a umožní mu kedykoľvek poslať dáta. Tým pádom počítač očakáva vstupné dáta a môže ich prijímať. Viac sa možno dočítať v [13].

Z toho ako USB funguje, vyplýva, že nie je možné aby Raspberry Pi Model B vystupoval ako Slave zariadenie a teda neumožňuje emuláciu USB klávesnice a myši. Avšak pomocou Raspberry Pi by bolo pravdepodobne možné emulovať klávesnicu pomocou PS/2 portu a GPIO portom umiestneným na doske. Dnes (v roku 2019) však ide už o zastaralé rozhranie, s ktorým sa môžeme stretnúť skôr u starších počítačov. V novších počítačoch bol tento port nahradený USB portom. Z tohto hľadiska nebola ďalej zamýšľaná emulácia PS/2 klávesnica, hlavne pre zastaranosť konektora a pre pravdepodobné obmedzenia vzniknuté počas vývoja.

Pre emuláciu USB klávesnice a myši je použitý mikrokontrolér Arduino Leonardo, ktorý dokáže fungovať ako Slave zariadenie a preto je možné vďaka nemu emulovať klávesnicu a myš.

2.9 Arduino

Arduino je open-source nízko nákladová platforma založená na mikroprocesoru ATmega. Vytvorené bolo v Taliansku zo snahou podporiť výučbu informatiky na školách a zoznámiť študentov s tým, ako sú pomocou počítačov riadené rôzne zariadenia, ako napríklad mikrovlnná trúba, automatická pračka a iné. Vďaka tomu, že Arduino je open-source platforma, existuje na trhu množstvo lacných čínskych klonov, ako je napr. Funduino, Femtoduino, FreeDuino a ďalšie. Všetky zariadenia majú v podstate rovnaký základ, ale líšia sa v rozložení portov na doske.

Spôčiatku môže Arduino pripomínať Raspberry Pi, avšak odlišnosť je pomerne veľká. Zatiaľ čo Raspberry Pi môže byť použité ako náhrada stolného počítača, u Arduina to možné nie je. Nie je možné na Arduino inštalovať napríklad Linux. Nemožno ku nemu priamo pripojiť monitor, klávesnicu ani myš. Na druhej strane je ale pripravené na pripojenie LED diód, servomotorov, senzorov, osvetlenia atď. Riadiaci program pre Arduino sa vyvíja zvlášť na stolnom počítači a následne je do Arduina nahraný. Vo vnútri Arduina je tento program spustený a typicky obsahuje slučku, ktorá sa stále opakuje. Arduino neustále zisťuje stav svojho okolia a reaguje na jeho zmeny. Má veľmi nízku spotrebu, je možné ho napájať len batériou a preto sa hodí napríklad pre riadenia dronov, robotov a podobne.

Dosky Arduino obsahujú 8 bitové mikroprocesory z rodiny AVR od firmy Atmel. Použité sú čipy ATmega8, ATmega168, ATmega328, ATmega1280 a ATmega2560. Arduino Due obsahuje 32-bitový ARM procesor Atmel SAM3X8E. Každá doska má väčšinou vstupno-výstupné piny prístupné cez štandardizované päťice, do ktorých sa jednoducho pripojujú ďalšie obvody, ktoré sa volajú Shields. Na doskách sa ďalej nachádza niekoľko diód, resetovacie tlačidlo, konektor pre ICSP programovanie, napájací konektor, oscilátor a obvod pre komunikáciu po USB. Základná verzia Arduina, obsahuje celkom 14 vstupno-výstupných digitálnych pinov a 6 analógových pinov. Šesť digitálnych pinov je možné použiť na softwarovo riadený PWM výstup. Hoci je Arduino pripojené k počítaču pomocou USB, sériová komunikácia cez linku RS-232 je simulovaná softvérovou. Viac sa možno dočítať v [14].

2.9.1 Bootloader

Hlavný mikrokontrolér, ktorý je užívateľsky programovateľný, má Bootloader. Je to vlastne kód, ktorý sa po spustení Arduina postará o základné nastavenie mikrokontroléru, ako sú interné časovače, nastavenie rozhrania UART a ďalšie. Uľahčuje tak užívateľovi prácu s Arduino a jeho použitie v projektoch, pretože nie je nutnosť mať hardvérové znalosti a starať sa tak o nastavenie mikrokontroléra. Programátorovi tak stačí iba napísať program v C/C++ a o ostatné sa postará Bootloader.

2.9.2 Arduino Shields

Pre Arduino existujú rozširovacie moduly, ktoré sa nazývajú Shields. Tieto Shields sa zapájajú do pinov na doske Arduina. Pomocou týchto modulov môžeme získať rôzne rozšírenia

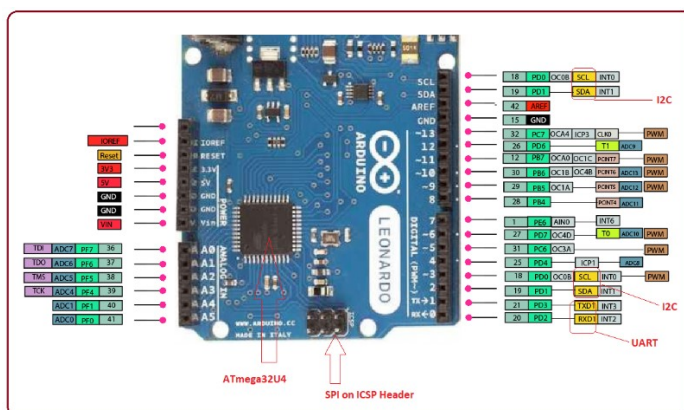
pre Arduino. Niektoré Shields používajú všetky dostupné piny na doske, iné iba niektoré. Podľa druhu modulu potom prebieha komunikácia s Arduino, buď po SPI, I2C alebo sériovej linky.

Jedným z najznámejších modulov je Ethernet Shield, ktorý umožňuje ku Arduino pripojiť sieťový kábel vďaka RJ-45 konektoru, okrem RJ-45 konektoru obsahuje slot pre pamäťovú kartu, resetovacie tlačidlo, indikačné LED diódy. Alternatívou je Wi-Fi Shield, ktorý obsahuje štandard 802.11b a 802.11g so zabezpečením WEP a WPA2. Ďalej to je napríklad GPS Shield, pomocou ktorého je okrem aktuálnej polohy možné zistiť nadmorskú výšku a rýchlosť.

2.9.3 Arduino Leonardo

Arduino Leonardo obsahuje mikrokontrolér ATmega32u4. Na doske sa nachádza 20 digitálnych vstupno-výstupných pinov. Z toho 7 pinov môže byť použitých ako PWM výstup a 12 ako analógové vstupy. Ďalej obsahuje 16MHz kryštálový mikroprocesor, microUSB, napájací konektor, ICSP konektor a resetovacie tlačidlo. Viac sa možno dočítať vo [15].

Hlavnou výhodou Leonarda je, že obsahuje mikroprocesor ATmega32u4, ktorý podporuje USB komunikáciu. To umožňuje, aby Leonardo mohlo na pripojenom počítači vystupovať ako klávesnica a myš. Práve preto bolo Leonardo vybrané pre výsledné riešenie.



Obrázok 2.10: Arduino Leonardo

Zdroj: The Engineering Projects, rok 2019

Hlavné rozdiely od Arduino Uno:

- Jeden procesor pre náčrty a USB komunikáciu: Týmto je umožnená väčšia flexibilita pri komunikácii s Počítačom a znižujú sa tak náklady na spotrebu.
- Resetovanie Sériovej linky: Sériový port u Leonarda je virtuálny, čo znamená, že pri každom resetovaní dosky dôjde ku zresetovaniu virtuálnej sériovej linky cez USB.
- Emulácia Klávesnice a Myši: Vďaka procesoru 32u4, je možné emulovať HID⁵ zariadenia klávesnicu a myš.

⁵ Human interface device

3 Návrh riešenia

V tejto kapitole sa venujem návrhu riešenia. Aby výsledné riešenie bolo čo najkvalitnejšie, je žiadúce si pred implementáciou ujasniť požiadavky, princípy fungovania a výber technológií.

3.1 Analýza požadovanej funkcionality

Predpokladaná funkcionality riešenia bude umožnenie pomocou jednoduchého webového rozhrania, vzdialene ovládať iné zariadenie. Výsledné riešenie by malo byť prehľadné, intuitívne a ľahko ovládateľné. Riešenie bude obsahovať samostatný mikropočítač, ktorý bude pripojený ku vzdialenému zariadeniu a webové rozhranie pomocou, ktorého bude možné sa pripojiť na mikropočítač a vďaka nemu ovládať vzdialené zariadenie.

Predpokladaná funkcionality webového rozhrania:

- Zobrazenie pripojených klientov
- Umožnenie pripojiť sa ku klientom
- Ovládať pripojeného klienta
 - Emulácia klávesnice
 - Emulácia myši
 - Zobrazenie kamerového vstupu

Predpokladaná funkcionality mikropočítača:

- Schopnosť automaticky sa pripojiť k serveru
- Schopnosť emulovať klávesnicu a myš
- Schopnosť nahrávať displej pomocou kamery

3.2 Princíp komunikácie

Keďže pristupujeme k ovládaniu vzdialeného počítača s kamerovým vstupom, je veľmi nebezpečné túto komunikáciu prevádzať bez akéhokoľvek zabezpečenia. Potenciálny útočník by mohol získať prístup k ovládaniu počítača, čo je nežiadúce. Preto je nutné komunikáciu zabezpečiť. O zabezpečenie komunikácie a pripojenie jednotlivých Raspberry Pi zariadení ku serveru sa postará OpenVPN server. Prístup do OpenVPN získajú iba vybrané zariadenia a klienti a tým zamedzíme prístup potenciálne nežiadúcim osobám.

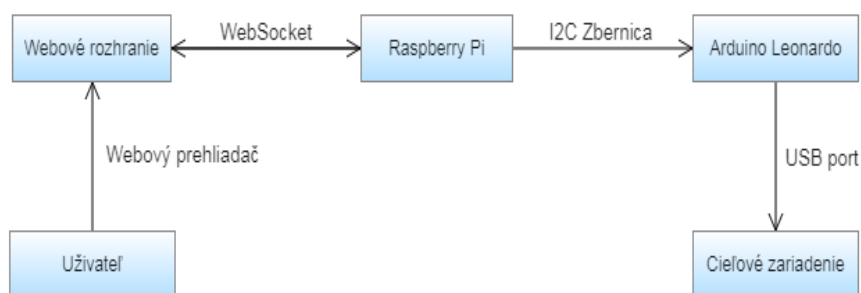


Obrázok 3.1: Schéma našej OpenVPN siete

Zdroj: Vlastné spracovanie autora.

Komunikáciu v rámci OpenVPN siete môžeme rozdeliť na tri dôležité časti, a to:

- Komunikácia Webového rozhrania s Raspberry Pi
- Komunikácia Raspberry Pi s Arduino
- Komunikácia Arduina s cieľovým počítačom



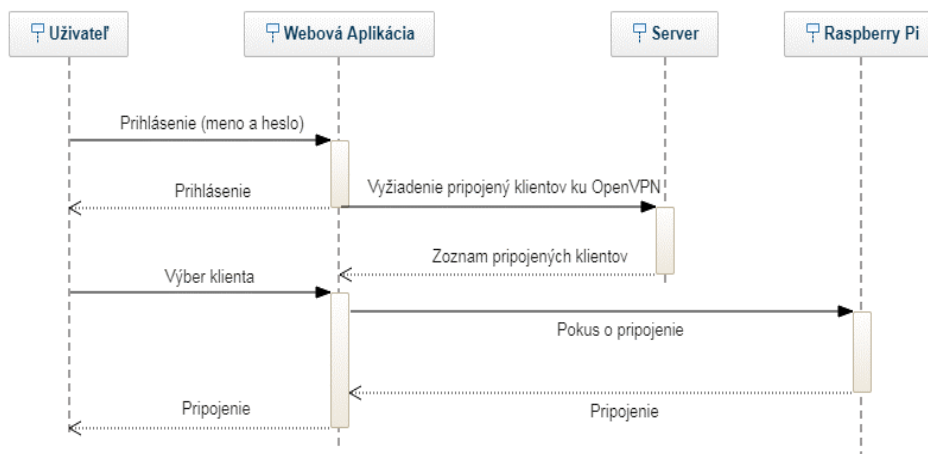
Obrázok 3.2: Schéma komunikácie medzi jednotlivými časťami riešenia

Zdroj: Vlastné spracovanie autora

Na obrázku 3.2 môžeme vidieť, celkový priebeh komunikácie s použitými technológiami medzi jednotlivými komunikačnými bodmi. Môžeme vidieť, že užívateľ na Webové rozhranie bude pristupovať cez klasický webový prehliadač. Pre umožnenie komunikácie z Webového rozhrania na Raspberry Pi bude použitý protokol WebSocket, ďalej tieto dáta bude Raspberry Pi preposielať na Arduino pomocou I2C zbernice. Poslednou časťou je pripojenie Arduina ku cieľovému zariadeniu pomocou USB portu, čo umožní emuláciu klávesnice a myši.

3.3 Webová aplikácia

Webová aplikácia sa skladá celkom z troch obrazoviek, a to z obrazovky pre prihlásenie, obrazovky zobrazujúca zoznam pripojených klientov a obrazovky slúžiacej ako interface pre ovládanie vzdialeného zariadenia. Navigácia po aplikácii je jednoduchá a pri tomto malom množstve obrazoviek sa užívateľ nestratí.



Obrázok 3.3: Diagram aktivít - priebeh fungovania webovej aplikácie

Zdroj: Vlastné spracovanie autora

3.3.1 Použitý framework

Pre vytvorenie webovej aplikácie bol zvolený AngularJS framework⁶, ktorý je vhodný na tvorbu SPA⁷ aplikácií (Single-page application). Hlavnou výhodou oproti klasickej HTML⁸ stránky je rýchla odozva. Po prvotnom načítaní stránky sa už nestahuje vždy celý HTML kód, ale iba dáta, ktoré sa menia. Okrem dátovej úspory nemusí prehliadač neustále prekresľovať celú stránku, ale iba zmenené časti. Poslednou výhodou SPA je relatívna jednoduchosť na tvorbu, pretože sa logika aplikácie vytvára iba v JavaScripte⁹.

3.3.2 AngularJS

AngularJS je JavaScriptový webový framework, ktorý v roku 2009 vytvorila spoločnosť Google. Zameriava sa na tvorbu SPA tvorených HTML kódom. Do HTML kódu sa vkladajú formátovacie značky, ktoré určujú aké dáta sa kam vložia. Obsahuje two-way data binding a má veľa ďalších užitočných vlastností vďaka čomu sa kód stáva prehľadný, jednoduchý a krátky.

⁶ Aplikačný rámec

⁷ Druh webovej stránky

⁸ Hypertextový značkový jazyk slúžiaci na vytváranie webových stránok

⁹ Skriptovací programovací jazyk

Two-way data binding (alebo dvojcestná synchronizácia dát) rieši synchronizáciu stavov medzi modelom a view¹⁰. Pre pochopenie ako to celé funguje, je potreba si predstaviť jednoduchú aplikáciu, v ktorej máme jedno textové pole a pod ním chceme vypísať čo užívateľ do poľa napísal. Vo chvíli kedy užívateľ napíše nejaký reťazec do poľa, sa tento text preniesie do modelu (to je prvá cesta) a následne sa z modelu prepíše do view (to je druhá cesta) bez nutnosti aktualizovať stránku. Viac sa možno dočítať v [16].

3.3.3 Grafické užívateľské rozhranie (GUI)

Pre jednotný štýl webovej aplikácie a príjemné užívateľské rozhranie bola použitá UI knižnica pre AngularJS. Použitá knižnica využíva vizuálne prvky Material dizajnu od Googlu. Obsahuje sadu opakovane použiteľných, dobre testovaných a prístupných komponentov pre GUI. Vďaka UI knižnici je vytváranie užívateľského rozhrania jednoduché a napomáha vzniknúť prehľadnému a funkčnému užívateľskému prostrediu.

3.3.4 Navigačná lišta

Nachádza sa v hornej časti stránky a je zobrazená na každej obrazovke. V pravom hornom rohu sa nachádza meno prihláseného užívateľa. Rozmiestnenie niektorých tlačidiel na navigačnej lište sa však mení v závislosti, od ktorej obrazovky sa užívateľ nachádza. V pravom hornom rohu na obrazovke pripojených klientov sa nachádza tlačidlo pre znovu načítanie pripojených klientov. V ľavom hornom rohu na obrazovke slúžiacej ako interface pre ovládanie sa nachádza tlačidlo pre odpojenie a na návrat na obrazovku pripojených klientov.

3.3.5 Prihlasovacia obrazovka

Pred prístupom do aplikácie je nutné preukázať svoju totožnosť, aby bol zamedzený prístup neoprávneným užívateľom. Pri načítaní webovej aplikácie sa zobrazí prihlasovacia obrazovka, kam užívateľ zadá prihlasovanie údaje. Po úspešnom prihlásení bude užívateľ presmerovaný na obrazovku pripojených klientov.

3.3.6 Zoznam pripojených klientov

Po úspešnom prihlásení užívateľa bude Webovou aplikáciou pomocou JavaScriptu na pozadí automaticky načítaný súbor s názvom: openvpn-status.log, v tomto súbore OpenVPN server uchováva zoznam pripojených klientov. Tento zoznam bude spracovaný JavaScriptom a užívateľovi sa zobrazia jednotliví pripojení klienti v podobe dlaždíc v GUI. Jednotlivé dlaždice budú obsahovať informácie o každom klientovi (napr. Názov, IP adresa a ďalšie).

Na každej dlaždici bude tlačidlo slúžiace na pripojenie ku vybranému klientovi, po stlačení tohto tlačidla bude užívateľ presmerovaný na obrazovku ktorá umožňuje vzdialené ovládanie klienta.

¹⁰ Je označenie časti, ktorá je zobrazená užívateľovi

3.3.7 Interface ovládanie klienta

Po pripojení na vybraného klienta bude užívateľ presmerovaný na obrazovku slúžiacu na ovládanie vzdialeného zariadenia. Na danej obrazovke bude zobrazený stream z kamery, ktorá bude nahrávať displej zariadenia. Ďalej tu budú ovládacie prvky na ovládanie klávesnice a myši na vzdialenom zariadení.

3.4 Aplikácia pre Raspberry Pi

Aplikácia pre Raspberry Pi bude spracovaná pomocou jazyka Java. V podstate sa bude jednať o medzičlánok medzi webovým rozhraním a Arduino. V jednoduchosti povedané, bude preposielať užívateľský vstup z webového rozhrania pre myš a klávesnicu na Arduino, ktoré tento vstup bude emulovať na koncovom zariadení.

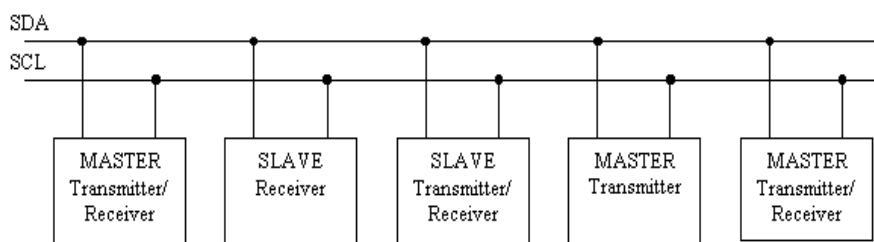
- Komunikácia Webového rozhrania s Raspberry Pi bude pomocou WebSocket
- Komunikácia Raspberry Pi s Arduino Leonardom bude pomocou I2C zbernice

Raspberry Pi zabezpečuje zobrazenie displeja cieľového zariadenia cez kameru umiestnenú na Raspberry Pi. Na tento účel bude použitý program Motion, ktorý umožňuje streamovať Raspberry Pi video z kamery po lokálnej sieti.

3.4.1 I2C zbernica

Zbernica I2C je dvojvodičové dátové spojenie medzi jedným alebo viacerými procesormi Master a špeciálnymi periférnymi súčiastkami Slaves. Všetky súčiastky sú pripojené do rovnakej zbernice a sú cielene vybrané svojimi adresami. Adresy a dáta sú prenášané rovnakými vodičmi. Zbernica umožňuje veľmi jednoduché prepojenie medzi niekoľkými integrovanými obvodmi a bezproblémové dodatočné rozširovanie.

I2C zbernica používa sériovú dátovú linku SDA (Synchronous Data), ktorá slúži na prenos dát a linku hodinového signálu SCL (Synchronous Clock), ktorá slúži na prenášanie spoločného hodinového signálu. Pri použití týchto liniek je nutné taktiež pridať spoločnú signálovú zem (GND).



Obrázok 3.4: Princíp prepojenia I2C zbernice

Zdroj: DH servis, rok 2019

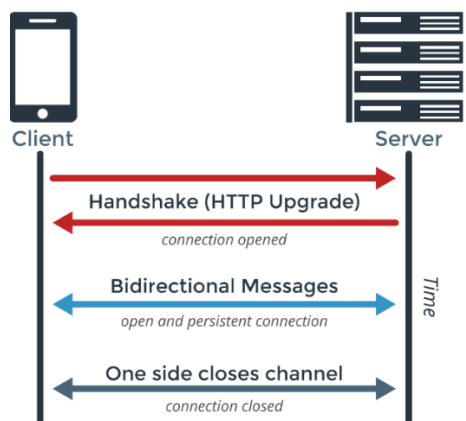
Obrázok 3.4 zobrazuje princíp prepojenia zbernice. Aby nevznikali kolízie na zbernici, môže iba jedno zariadenie figurovať ako typ Master a všetky ostatné zariadenie fungujú ako Slave a zbernicu nemôžu riadiť. Master zariadenie ovláda ostatné zariadenia a generuje na vodiči SCL hodinový signál a skrz SDA prebieha výmena dát so zariadeniami Slave.

- Master - Zahajuje komunikáciu a posíla alebo číta dáta z podriadených (Slave) zariadení. Master pozná adresu každého zariadenia a keď potrebuje osloviť konkrétne zariadenie, urobí to pomocou adresy.
- Slave - Zariadenie čaká na svoje oslovenie a keď je oslovené Master zariadením tak si buď len prečíta aké údaje prijalo alebo môže aj odpovedať svojimi údajmi.

Podľa špecifikácie je možné pripojiť až 128 Slave zariadení. Viac na [17].

3.4.2 WebSocket

WebSocket je počítačový komunikačný protokol, poskytuje obojsmernú komunikáciu cez jedno TCP¹¹ pripojenie. WebSocket je navrhnutý tak aby mohol fungovať vo webovom prehliadači a na webových serveroch. Je to nezávislý protokol založený na TCP. Protokol WebSocket umožňuje interakciu medzi prehliadačom a webovým serverom. Uľahčuje real-time¹² prenos dát zo serveru a na server. Vďaka tomu poskytuje štandardizovaný spôsob pre odoslanie obsahu zo serveru do prehliadača, bez toho aby to bolo na požiadavku klienta a umožňuje predávanie správ tam a späť, zatiaľ čo je udržiavané otvorené spojenie. Týmto spôsobom môže prebiehať obojsmerná komunikácia medzi prehliadačom a serverom. Komunikácia prebieha cez TCP port číslo 80, v prípade TLS-šifrovania spojenia cez port 443. Protokol WebSocket je v súčasnej dobe podporovaný väčšinou najpoužívanejších webových prehliadačov. WebSocket vyžaduje podporu tiež na strane serveru. Viac sa možno dočítať v [18].



Obrázok 3.5: *Priebeh komunikácie cez WebSocket*

Zdroj: Medium, rok 2019

¹¹ Protokol riadenia prenosu

¹² V reálnom čase

3.4.3 Program Motion

Program Motion je veľmi dobre konfigurovateľný program. Spolupracuje s veľkým množstvom kamier. Je používaný na monitorovanie bezpečnostných kamier, sledovanie vtákov, kontrolu domácich malíčkov a vytvárania časozberných záznamov. Viac na [19].

Hlavné funkcie Motion:

- Vytvorenie videa alebo fotiek pri nejakej aktivite v obraze
- Možnosť sledovania živého vysielania s kamery
- Spúšťanie skriptov po zaznamenaní aktivity
- Logovanie aktivity
- Pohybovú detekciu

Pre výsledný projekt bude využitá práve funkcia živého vysielania.

3.5 Program na emuláciu klávesnice a myši

Emuláciu klávesnice a myši bude zabezpečovať mikrokontrolér Arduino Leonardo. Na emuláciu budú použité knižnice Keyboard a Mouse, ktoré sú oficiálnym produktom Arduina. Príkazy bude prijímať od mikropočítača Raspberry Pi pomocou I2C zbernice a ku cieľovému zariadeniu bude pripojené pomocou USB portu. Program bude napísaný v jazyku C. Viac na [20].

3.6 Krabíčka pre zariadenie

Môže sa zdať ako nepodstatná vec, ale pre využitie v reálnom prostredí je krabíčka nutnosť. Aby bola umožnená jednoduchá manipulácia, prenositeľnosť a možnosť nastavenia kamery na displej ovládaného počítača. Na tento účel bude vytvorený 3D model krabíčky a následne bude krabíčka vytlačená na 3D tlačiarňi.

3.7 Zautomatizovanie Raspberry Pi

Poslednou navrhovanou a dôležitou časťou je, aby Raspberry Pi dokázalo pracovať úplne samostatne. To znamená, aby sa automaticky dokázala pripojiť na OpenVPN server pri pripojení ku vzdialenému počítaču. Tak isto umožnenie pripojenia do neznámej Wi-Fi siete, pomocou Bash Skriptu¹³ a USB kľúča, na ktorom v textovej podobe bude umiestnené SSID¹⁴ Wi-Fi a heslo.

¹³ Je to skript napísaný pre príkazový riadok

¹⁴ Je identifikátor bezdrôtovej siete Wi-Fi

4 Konfigurácia USB modemu

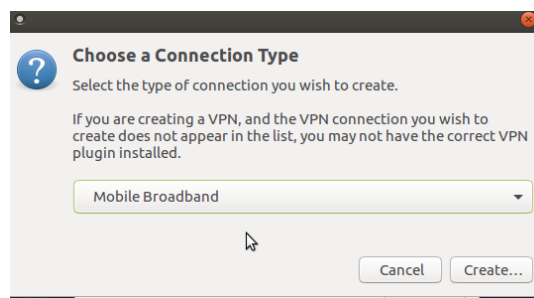
Nasledujúca kapitola sa zaoberá konfiguráciou USB modemu pre sprístupnenie internetu po mobilnej sieti.

4.1 Použitý modem

Ako testovací USB modem bol použitý model USB Modem Huawei K3765 3G. Jedná sa síce len o 3G modem, ale pre otestovanie a konfiguráciu úplne postačuje, pretože konfigurácia bude rovnaká aj pre iné modely USB modemov. Vybraný modem podporuje technológie EDGE, CDMA a 3G. V danej oblasti vyberie najrýchlejšie dostupné mobilné pripojenie a sám sa k nemu pripojí.

4.2 Konfigurácia

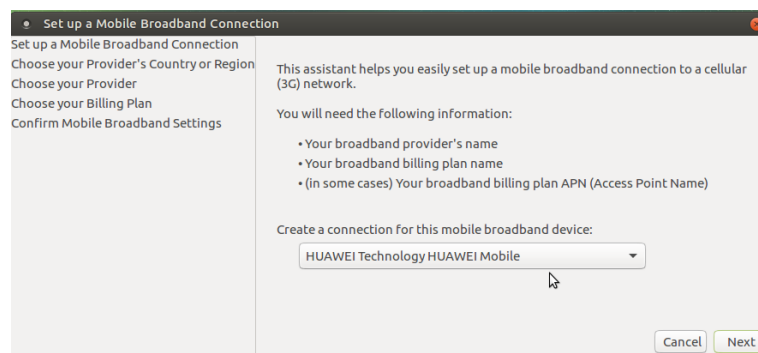
Pre uľahčenie konfigurácie bola konfigurácia prevádzaná v užívateľskom prostredí operačného systému. Po zapojení modemu do USB portu, prejdeme do nastavení siete a dáme tlačidlo pridať.



Obrázok 4.1: *Výber typ pripojenia*

Zdroj: Vlastné spracovanie autora

Ako typ pripojenia vyberieme Mobilné pripojenie. V sprievodcovi nastavenia mobilného pripojenia vyberieme náš pripojený modem.



Obrázok 4.2: *Výber modemu*

Zdroj: Vlastné spracovanie autora

Následne sa v sprievodcovi vyberá krajina alebo región, v tomto prípade sa vybrala Česká Republika. V ďalšom bode je na výber poskytovateľ, v tomto prípade sa vyberal T-Mobile. Posledným krokom bolo zadanie APN¹⁵. Týmto je modem úspešne nastavený. Ešte bolo na konci nutné editovať sieť a vybrať možnosť Pripojiť sa automaticky ak je sieť k dispozícii. Takto sa pri štarte, pri pripojenom modeme, Raspberry Pi automaticky pripojí na mobilné dáta.

Presné nastavenie APN pre operátora T-Mobile môžeme vidieť nižšie. Zdroj [21].

Name: T-Mobile CZ

APN: internet.t-mobile.cz

Proxy: <Not set>

Port: <Not set>

Username: <Not set>

Password: <Not set>

Server: <Not set>

MMSC: <Not set>

MMS proxy: <Not set>

MMS port: <Not set>

MCC: 230

MNC: 01

APN type: default

Number: *99#

¹⁵ Meno pristupovaného bodu

5 Inštalácia serverovej platformy s OpenVPN

Nasledujúca kapitola sa venuje konfigurácii serverovej časti, výberu operačného systému pre server a popise a konfigurácii OpenVPN serveru.

Ako server bolo zvolené cloudové riešenie. Výhodou VPS serveru sú nízke mesačné náklady, pri najnižšej možnej konfigurácii, ktorá je však dostačujúca. Hlavnou výhodou je verejná IP adresa a teda dostupnosť kedykoľvek a odkiaľkoľvek.

5.1 Operačný systém servera

Operačný systém bol zvolený Debian verzie 8 64bit. Debian je jeden z mnohých linuxových distribúcií. Patrí k jednej z najstarších do dnes (v roku 2019) vyvíjaných distribúcií, ktorú nevyvíja komerčný subjekt ale veľké množstvo dobrovoľníkov z celého sveta. Je to jedna z najrozšírenejších linuxových distribúcií na svete. Pre serverové inštalácie je obľúbený predovšetkým vďaka svojej stabilite a jednoduchšej údržbe. Vďaka prísnyim smerniciam na kvalitu balíkov, je ich možné aktualizovať medzi rôznymi vydaniaми bez nutnosti reštartovania počítača. Viac na [22].

Keďže bol použitý VPS server, tak samostatná inštalácia systému prebehla automaticky. Pri vytváraní cloudového servera bolo na výber niekoľko šablón operačných systémov. Posledným krokom bolo zadanie hesla pre správcu servera a server bol pripravený.

Manuálna inštalácia Debian-u však nie je vôbec zložitá, stačí postupovať podľa pokynov v sprievodcovi inštalácie.

5.2 OpenVPN

Keďže zariadenie pre vzdialenú správu bude umiestňované mimo lokálnu a zabezpečenú sieť je nutné zabezpečiť s ním komunikáciu. Pre tento účel bude použitý softvér pre vytvorenie VPN, čo je vlastne Virtuálna privátna sieť. VPN umožňuje komunikáciu medzi zariadeniami, ktoré sú umiestnené na rozdielnych miestach a v rozdielnych sieťach ako keby boli v jednej uzavretej privátnej sieti. Ako VPN softvér bolo zvolené open-source riešenie OpenVPN. Viac sa možno dočítať v [23].

5.2.1 Čo je to OpenVPN

OpenVPN je voľne dostupný softvér, ktorý má verejný kód. Slúži na vytvorenie šifrovaného VPN tunelu medzi zariadeniami. Využíva model klient-server. Výhodou je, že dokáže zaistiť priame spojenie medzi zariadeniami za NAT¹⁶ a to bez potreby NAT akokoľvek konfigurovať. OpenVPN je dostupný pod licenciou GNU, čo znamená, že sa jedná o slobodný softvér.

¹⁶ Slúži na preklad adres

Vďaka OpenVPN je možné vytvárať bezpečnú a ľahko konfigurovateľnú VPN sieť. Na zabezpečenie využíva protokol SSL/TLS. Na šifrovanie je možné použiť ľubovoľný šifrovací algoritmus, ktorý podporuje SSL knižnica. Štandardne využíva protokol UDP, avšak je možné použitie aj TCP. Celá komunikácia prebieha po jednom porte, čo umožňuje ľahko nakonfigurovať firewall, aby boli prepúšťané pakety len cez tento port.

OpenVPN komunikuje prostredníctvom TAP a TUN rozhrania. TAP a TUN sú v podstate virtuálne sieťové rozhrania, ktoré umožňujú aplikáciám v užívateľskom prostredí komunikovať so sieťovým rozhraním, bez nutnosti root¹⁷ oprávnenia. Každá aplikácia, ktorá používa sieťové rozhranie, má možnosť využiť TUN/TAP rozhranie VPN tunelu. Umožňuje vytvorenie IP tunelu (TUN) alebo Ethernetov konektivity (TAP). Od verzie OpenVPN 2.6 je plne podporovaná tiež IPv6.

OpenVPN má k dispozícii niekoľko možností overenia klientov:

- Overenie pomocou Zdieľaného kľúča.
- Overenie pomocou Certifikátu.
- Overenie pomocou Mena a hesla.

Najjednoduchšie riešenie je použitie zdieľaného kľúča, no bezpečnejšia cesta je využitie certifikátu alebo užívateľského mena a hesla, prípadne ich kombinácia.

5.2.2 Inštalácia a konfigurácia OpenVPN

Samostatnú inštaláciu a konfiguráciu OpenVPN serveru môžeme rozdeliť do niekoľkých bodov:

- Inštalácia OpenVPN a EasyRSA
- Konfigurácia EasyRSA a vytvorenie CA(certifikačnú autoritu)
- Certifikáty pre server a klientov
- Konfigurácia OpenVPN
- Štart OpenVPN servera

OpenVPN je dostupný pre inštaláciu z oficiálnych repozitárov pre Debian. Jednoducho ho môžeme nainštalovať pomocou nasledovného príkazu.

```
sudo apt-get install openvpn
```

Ďalší krok je inštalácia EasyRSA. EasyRSA uľahčuje tvorbu certifikátov. Generovanie kľúčov, žiadostí a certifikátov je samozrejme možné aj ručne pomocou OpenSSL, ale EasyRSA túto cestu výrazne zjednodušuje.

```
sudo apt-get install easy-rsa
```

Po nainštalovaní OpenVPN a EasyRSA prejdeme na vytvorenie certifikačnej autority (CA). Najskôr si však skopírujeme do vlastného adresára.

¹⁷ Je označenie pre super užívateľa v prostredí Linux

```
cp -r /usr/share/easy-rsa /home/user
```

Otvoríme súbor "vars" napríklad pomocou nano¹⁸ a niektoré premenné upravíme. Premenné v tomto súbore definujú parametre našej certifikačnej autority. Väčšinu môžeme však nechať v nezmenenom stave. Zásadné parametre sú uvedené nižšie.

```
export KEY_COUNTRY=""      # krajina
export KEY_PROVINCE=""     # provincia
export KEY_CITY=""         # mesto
export KEY_ORG=""          # organizácia
export KEY_EMAIL=""        # mail
export KEY_OU=""
```

Treba venovať pozornosť tiež parametrom "CA_EXPIRE" a "KEY_EXPIRE", ktoré definujú počet dní platnosti autority a jednotlivých certifikátov. Východzia hodnota je nastavená na 10 rokov.

Po týchto úpravách už je možné generovať kľúče a certifikáty. Pomocou nasledovného príkazu.

```
./build-ca
```

Týmto sa vytvorila certifikačná autorita a v adresári "keys" vznikli dva nové súbory "ca.key" s kľúčom a "ca.crt" s certifikátom. Teraz už len ostáva vygenerovanie certifikátov pre server a jednotlivých klientov. Ku generovaniu certifikátu stačí spustiť ďalší príkaz s parametrom, ktorý obsahuje unikátny názov severu.

```
./build-key-server NášServer
```

V adresári "keys" znova vzniknú dva nové súbory a to "NášServer.key" s kľúčom a "NášServer.crt" s certifikátom. Pre server je ďalej nutné vygenerovať parametre pre výmenu kľúčov algoritmom Diffie-Hellman¹⁹.

```
/build-dh
```

Po tomto sa v adresári "keys" vytvorí nový súbor "dh2048.pem". Ďalej je nutné pre každého klienta, ktorý sa bude chcieť pripojiť do našej VPN, vygenerovať samostatný certifikát.

```
./build-key MenoKlienta
```

Znovu sa v adresári "keys" vytvorí nové súbory "MenoKlienta.key" s kľúčom a "MenoKlienta.crt" s certifikátom.

¹⁸ Je malý a jednoduchý textový editor v prostredí Linux

¹⁹ Algoritmus, ktorý slúži k dohode spoločného kľúča po nezabezpečenom kanále.

Posledným krokom je vytvoriť konfiguračné súbory pre server a klientov. Na serveri musia byť prítomné nami vygenerované súbory z predchádzajúcej časti:

- dh2048.pem; ca.crt; NášServer.crt; NášServer.key

Pomocou svojho certifikátu bude server preukazovať klientom svoju identitu a ich identitu bude overovať pomocou certifikátu dôveryhodnej authority. U klienta musia byť prítomné súbory:

- ca.crt; klient.crt; klient.key

V konfiguračnom súbore pre server sa nachádzajú parametre, ako napríklad:

- aký protokol bude použitý
- cesta ku certifikátom
- názov sieťového rozhrania
- rozsahy IP adries udeľované klientom
- cesta k súboru o stave servera
- a iné

V konfiguračnom súbore pre klientov sa potom nachádza:

- názov servera
- sieťové rozhranie
- cesta ku certifikátom
- a iné

Vysvetlenie všetkých parametrov, ktoré môžu obsahovať jednotlivé konfiguračné súbory nájdeme v dokumentácii OpenVPN²⁰. Po vytvorení konfiguračných súborov už stačí len reštartovať server. Jednotlivé konfiguračné súbory sú priložené v prílohe.

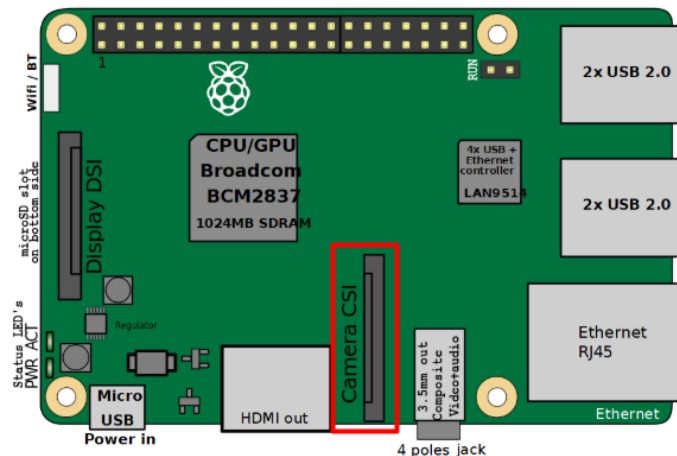
²⁰ <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/>

6 Inštalácia kamery s umožnením streamovania

Nasledujúca kapitola sa venuje inštalácii kamery na Raspberry Pi s možnosťou vzdialeného sledovania obrazu.

6.1 Inštalácia kamery

Prvým krokom je zapojenie kamerového modulu do dosky Raspberry Pi. Kamera sa pripája do CSI portu na doske.

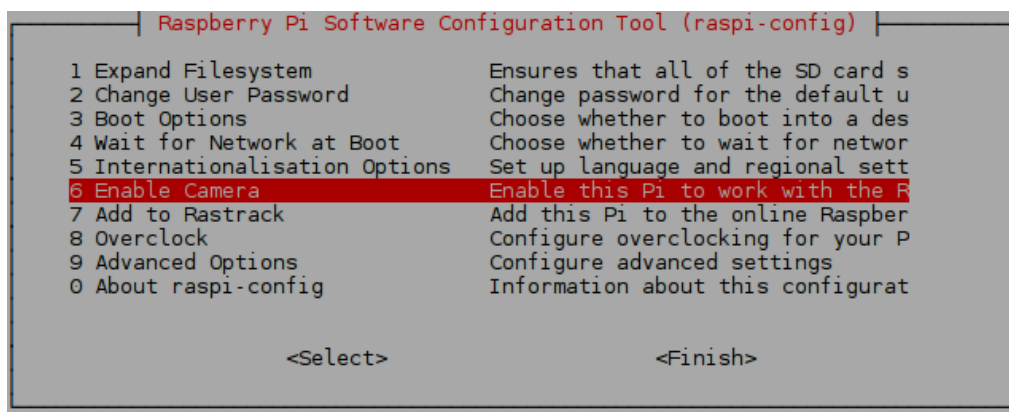


Obrázok 6.1: Vyznačený CSI port na doske Raspberry Pi 3

Zdroj: UniPi.technology, rok 2019

Po zapojení kamery do dosky, môžeme spustiť Raspberry Pi. V operačnom systéme následne kameru povolíme. Nasledujúcim príkazom sa dostaneme do konfigurácie Raspberry Pi.

```
sudo raspi-config
```



Obrázok 6.2: Zobrazenie konfiguračného nástroja raspi-config

Zdroj: Vlastné spracovanie autora

V konfiguračnom nástroji prejdeme na možnosť Kamera, kde ju aktivujeme. Nakoniec reštartujeme Raspberry Pi a kameru môžeme používať.

6.2 Program Motion - vzdialené sledovanie obrazu

Pre nainštalovanie Motion programu slúži nasledovný príkaz:

```
sudo apt-get install motion
```

Po nainštalovaní je nutné upraviť konfiguračný súbor. V konfiguračnom súbore, ktorý sa nachádza v `"/etc/motion/motion.conf"` vyhľadáme nasledujúce časti a upravíme ich:

```
# Povolenie štartu na pozadí
daemon on

# Nastavenie umiestnenia logovacieho súboru
logfile /var/log/motion/motion.log

# Povolenie streamu len cez localhost
stream_localhost off

# Zakázanie ukladania fotiek a videí
output_pictures off
ffmpeg_output_movies off

# Nastavenie snímkovej frekvencie
framerate 100

# Nastavenie rozlíšenia obrazu
width 640
height 480

# TCP/IP port pre stream
webcontrol_port 8080
```

Po nakonfigurovaní zostáva len spustenie Motion servisu. Následne môžeme pristupovať pomocou webového prehliadača na adrese `http://localhost:8080` ku streamu obrazu z kamery. Celý konfiguračný súbor je súčasťou prílohy.

7 Implementácia webového rozhrania pre serverovú platformu

Nasledujúca kapitola sa venuje implementácii webového rozhrania. Webové rozhranie obsahuje databázu pripojených klientov ku OpenVPN serveru. Umožňuje pripojenie k vybranému zariadeniu a následne ho pomocou webového rozhrania ovládať.

7.1 Vývojové prostredie WebStorm

Ako vývojové prostredie bolo použité IDE²¹ WebStorm od JetBrains. Patrí medzi najlepšie IDE pre moderný vývoj JavaScriptu. Podporuje technológie ako JavaScript, HTML a CSS, ale aj AngularJS, TypeScript, Node.js, Meteor, React, Cordova a ďalšie. Ponúka veľmi moderné a užívateľsky prívetivé rozhranie s mnohými funkciami, ktoré pomáhajú s písaním kódu.

Niektoré funkcie, ktoré ponúka WebStorm:

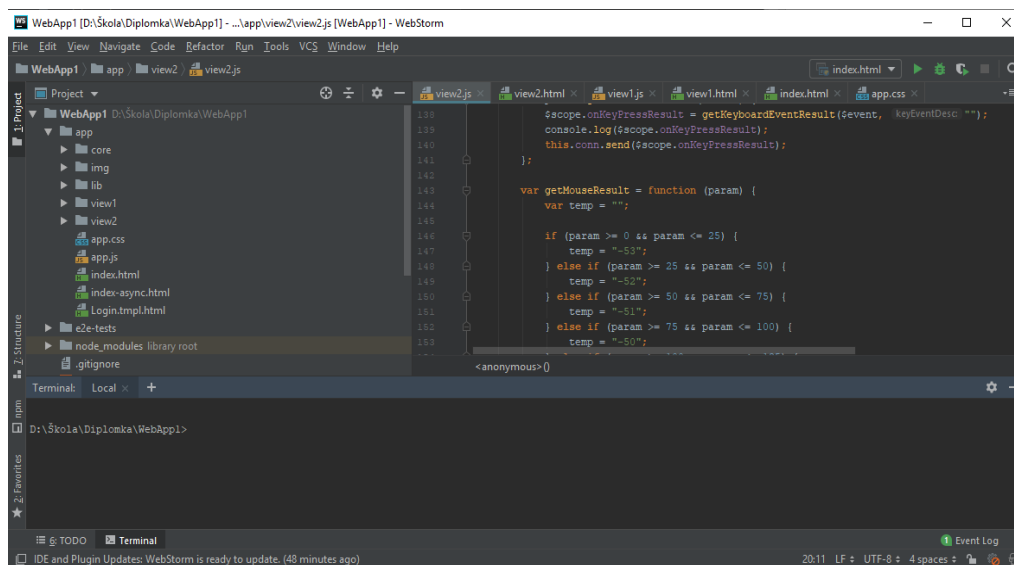
- Kompatibilita s Windows, Linux a Mac OS
- Inteligentné dokončovanie kódu
- Viac riadkové TODO²²
- Automatizované refaktorovanie kódu
- Debugger
- Kontrola syntaxe kódu
- Unit testovanie
- Integrácia GIT²³
- Nápovede k parametrom
- Integrácia VCS
- Prehľadná navigácia
- Možnosť pridania rôznych pluginov
- Integrovaná Windows konzola

Veľkou výhodou taktiež je, že dokáže vytvoriť základnú kostru webovej aplikácie pre vybranú technológiu. Základnú aplikáciu dokážeme vďaka WebStorm-u vytvoriť bez predošlých znalostí. Viac na [24].

²¹ Označenie pre vývojové prostredie

²² TODO slúži ako komentár v kóde, ktorý sa používa na zápis neskoršieho vylepšenia

²³ Je distribuovaný systém riadenia revízií



Obrázok 7.1: Ukážka WebStorm IDE

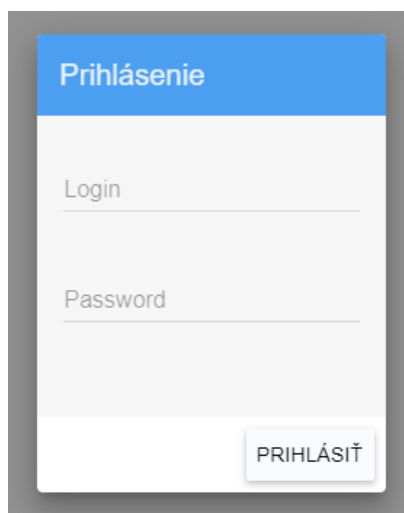
Zdroj: Vlastné spracovanie autora

7.2 Samotná webová aplikácia

Webová aplikácia je nasadená na našom VPS serveri a dostupná je iba v rámci OpenVPN siete. Je napísaná pomocou AngularJS. Po úspešnom prihlásení užívateľovi zobrazuje zoznam pripojených klientov.

7.2.1 Prihlásenie

Pred prístupom do aplikácie je nutné sa prihlásiť. Po zadaní mena a hesla užívateľom, sú tieto informácie overené. Následne je umožnený alebo zamedzený prístup do aplikácie.



Obrázok 7.2: Prihlasovacia obrazovka

Zdroj: Vlastné spracovanie autora

7.2.2 Získanie zoznamu pripojených klientov

Aby bolo možné zobrazit' pripojených klientov, je nutné požiadať server o súbor, ktorý tieto informácie obsahuje. Pre vyžiadanie informácií zo servera sa používa objekt XMLHttpRequest.

```
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        // Spracovanie súboru
    }
};
xhttp.open("GET", "filename", true);
xhttp.send();
```

Príklad implementácie XMLHttpRequestu.

Informácie o pripojených klientov ukladá OpenVPN server do súboru "openvpn-status.log", ktorý je v textovej podobe. Keďže tento súbor je uložený v adresári "/etc", ktorý obsahuje konfigurácie systému, nie je možné ho vyžiadať z aplikácie. Aby bol tento súbor dostupný pre aplikáciu, je treba vytvoriť odkaz na súbor, ktorý bude umiestnený v adresári našej webovej aplikácie. Vytvoríme ho pomocou nasledovného príkazu:

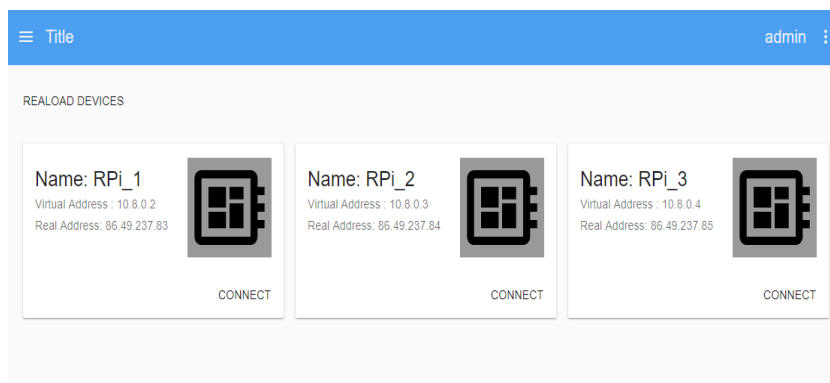
```
ln /etc/openvpn/openvpn-status.log /var/www/html/openvpn-
status.log
```

Príklad obsahu súboru:

```
OpenVPN CLIENT LIST
Updated,Sun Apr 28 21:49:55 2019
Common Name,Real Address,Bytes Received,Bytes Sent,Connected
Since
Mobile_1,86.49.224.100:52442,28836,60685,Sun Apr 28 21:49:23
2019
ROUTING TABLE
Virtual Address,Common Name,Real Address,Last Ref
10.8.0.3,Mobile_1,86.49.224.100:52442,Sun Apr 28 21:49:46 2019
GLOBAL STATS
Max bcast/mcast queue length,0
END
```


V "ROUTING TABLE" sú umiestnené informácie, ktoré sú spracované JavaScriptovým kódom.

Najdôležitejšia je Virtuálna adresa, ktorú priradzuje OpenVPN server. Na tejto adrese prebieha pripojenie Webového rozhrania na klienta a následne je umožnené jeho ovládanie. Ďalšou dôležitou informáciou je názov, podľa ktorého vieme identifikovať klienta a tak sa rozhodnúť ku komu sa pripojiť.



Obrázok 7.5: Zobrazenie klientov v aplikácii

Zdroj: Vlastné spracovanie autora

Každá jedna dlaždica v užívateľskom GUI predstavuje jedného klienta. Pomocou tlačidla "Connect" prebieha pripojenie na klienta.

7.2.3 Ovládanie klienta

Pripojenie na klienta prebieha pomocou WebSocket. WebSocket využíva klient-server model. V tomto prípade webové rozhranie vystupuje v roli klienta, ktoré sa pripája na server, ktorý beží na Raspberry Pi. Server beží na IP adrese pridelenej OpenVPN (Virtuálna adresa) serverom a na porte 80.

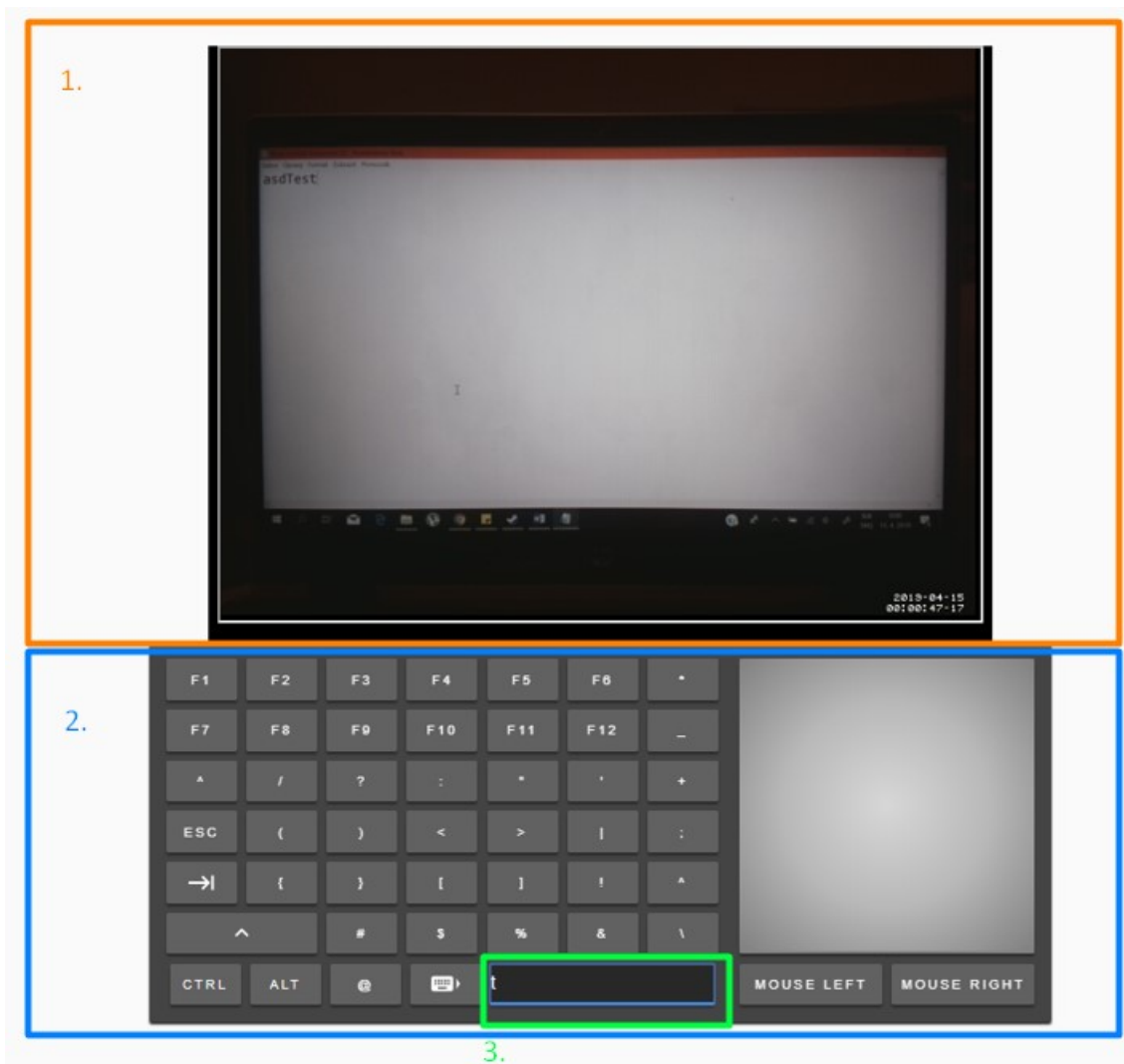
```
$scope.conn = new WebSocket("ws://" + "10.8.0.2" + ":80");  
$scope.conn.onopen = function (event) {  
};  
$scope.conn.onmessage = function (e) {  
};
```

Príklad implementácie WebSocket klienta.

Po úspešnom pripojení na klienta (Raspberry Pi) je možné plnohodnotne ovládať vzdialené zariadenie pomocou klávesnice a myši.

Obrazovka pre ovládanie je rozdelená na dve časti:

- Zobrazenie vstupu s kamery
- Ovládacie prvky predstavujúce klávesnicu a myš.



Obrázok 7.7: Ovládacie prvky

Zdroj: Vlastné spracovanie autora

Na obrázku 7.7 môžeme vidieť niekoľko ovládacích prvkov. Pre lepšie pochopenie významu boli na obrázku ohraničené jednotlivé časti, ktoré sú:

- 1. Zobrazuje obraz z kamery, ktorá nahráva displej vzdialeného zariadenia
- 2. Obsahuje klávesy špeciálnych znakov a touchpad s ľavým a pravým klikom myši
- 3. Umožňuje klávesový vstup ktorý sa v reálnom čase píše na vzdialenom počítači

Pre zobrazenie živého prenosu kamery z programu **Motion** je využívaný HTML tag **<iframe>**. Pre klávesový vstup je využívaných niekoľko eventov: **onKeyDown(\$event)**, **onKeyUp(\$event)** a **onKeyPress(\$event)**. Na zachytávanie pohybu myši sú využívané eventy: **moveHandler(\$event)**, **mouseupHandler(\$event)** a **mousedownHandler(\$event)**. Celé výsledné webové rozhranie je priložené v prílohe.

7.3 Nasadenie na VPS Server

Aby Webové rozhranie mohlo fungovať na našom VPS serveri, je nutné nainštalovať HTTP server. Ako HTTP server bol použitý Apache2 Web Server.

7.3.1 Čo je to Apache2 Web Server

Apache2 je open-source projekt. Je to najčastejšie používaný webový server v systéme Linux. Webové servery slúžia na poskytovanie webových stránok požadovaných užívateľskými počítačmi. Klienti požadujú a prezerajú webové stránky pomocou webového prehliadača. Viac na [25].

7.3.2 Konfigurácia Apache2 Web Serveru

Apache2 Web Server môžeme nainštalovať pomocou nasledovného príkazu:

```
sudo apt install apache2
```

Príklad môjho konfiguračného súboru, ktorý sa nachádza v adresári "/etc/apache2/sites-available" vyzerá nasledovne:

```
<VirtualHost 10.8.0.1:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory />
        Options FollowSymLinks Indexes
        AllowOverride None
    </Directory>
    <Directory /var/www/html>
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1
        Allow from 10.8
    </Directory>
</VirtualHost>
```

V konfiguračnom súbore je nastavená IP adresa, na ktorej funguje Webové rozhranie v rámci OpenVPN siete. V tomto prípade 10.8.0.1. Ďalej je nastavený adresár našej stránky "/var/www/html" a poslednou vecou sú nastavené prístupové práva na stránku. Povoľený prístup je len z localhostu a našej OpenVPN siete.

8 Implementácia Raspberry Pi aplikácie

Nasledujúca kapitola sa venuje implementácii aplikácie pre Raspberry Pi. Aplikácia pre Raspberry Pi obsahuje WebSocket server, na ktorý sa pripája Webové rozhranie. Ďalej pomocou I2C zbernice preposiela príkazy na Arduino Leonardo. Napísaná je v jazyku Java.

8.1 Vývojové prostredie IntelliJ IDEA

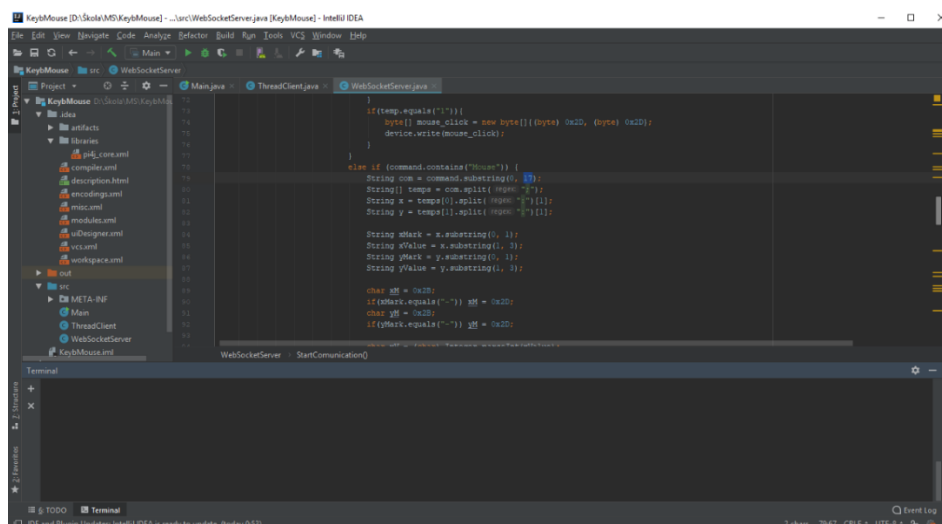
Ako vývojové prostredie bolo použité IDE IntelliJ IDEA od JetBrains. Na tomto vývojom prostredí je postavené aj Android Studio. K dispozícii je bezplatná komunitná verzia alebo platená verzia, ktorá obsahuje všetky nástroje. Viac na [26].

Niektoré funkcie, ktoré ponúka IntelliJ IDEA:

- Rozsiahli databázový editor a UML²⁴ dizajnér
- Inteligentný editor pre HTML, CSS²⁵ a Java
- Integrácia GIT
- Podporuje JavaEE, Spring, Hibernate, Google App Engine a ďalšie
- Integrovaná kontrola verzií
- Debugger
- Kontrola syntaxe kódu
- Nápovede k parametrom
- Prehľadná navigácia
- Možnosť pridania rôznych pluginov
- Integrovaná Windows konzola
- Revízie

²⁴ Grafický jazyk na vizualizáciu, špecifikáciu, navrhovanie a dokumentáciu programových systémov

²⁵ Kaskádové štýly, slúžiace na vizuálne formátovanie internetových dokumentov



Obrázok 8.1: Vývojové prostredie IntelliJ IDEA

Zdroj: Vlastné spracovanie autora

8.2 Samostatná aplikácia

Aplikácia beží na zariadení Raspberry Pi, je napísaná v Jave. Obsahuje WebSocket server, ktorý slúži na komunikáciu s Webovým rozhraním. Ďalej pomocou I2C zbernice prebieha komunikácia s Arduino Leonardo.

8.2.1 Java knižnica pre I2C zbernicu

Pre možnosť využívať GPIO porty u Raspberry Pi pomocou Javy, ktorá ju v základe nepodporuje, bolo nutné využiť Java knižnicu. Použitá bola Pi4J projekt. Táto knižnica umožňuje využívať pomocou Javy GPIO porty u platformy Raspberry Pi. Projekt je stále vo vývoji, ale už niekoľko rokov je veľmi stabilná. Sprístupnená je pod LGPL licenciou, ktorá umožňuje spájanie s neslobodným kódom. Podporované je veľké množstvo Raspberry Pi dosiek. Aktuálna verzia je 1.2 (v roku 2019). Viac na [28].

Vlastnosti:

- Prístup k systémovým a sieťovým informáciám
- Umožňuje rozšíriť GPIO prostredníctvom rozširujúcich dosiek
- SPI komunikácia
- I2C komunikácia
- Komunikácia cez sériovú linku RS232
- Automatické spúšťanie na základe zmien stavov pinov
- Automatické nastavenie GPIO stavov na ukončenie programu
- Počúvanie zmien GPIO pinov
- Čítanie stavov GPIO pinov
- Ovládanie a zápis stavov GPIO pinov
- Nastavenie GPIO pinov

Súčasne sa na oficiálnych stránkach projektu nachádza veľké množstvo príkladov použitia danej knižnice. Vďaka týmto príkladom je integrácia do projektov veľmi jednoduchá.

8.2.2 Komunikácia medzi Webovým rozhraním a Raspberry Pi

Pre komunikáciu s Webovým rozhraním sa využíva WebSocket. V tomto prípade Raspberry Pi vystupuje v roli serveru a čaká na pripojenie klienta. Server beží na localhoste a je dostupný na IP adrese, ktorá bola pridelená OpenVPN server. Port je číslo 80.

```
ServerSocket server = new ServerSocket(80);  
System.out.println("Server has started.");  
Socket client = server.accept();
```

Príklad implementácie WebSocket severu

Klient pri nadviazaní spojenia, posíla na server štandardnú HTTP požiadavku, ktorej hlavička vyzerá nasledovne:

```
GET /example HTTP/1.1  
Host: example.com:8000  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==  
Sec-WebSocket-Version: 13
```

Server po obdržaní požiadavky na pripojenie posíla späť špeciálnu odpoveď, ktorá indikuje, že protokol sa zmení z HTTP na WebSocket. Hlavička vyzerá nasledovne:

```
HTTP/1.1 101 Switching Protocols  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
```

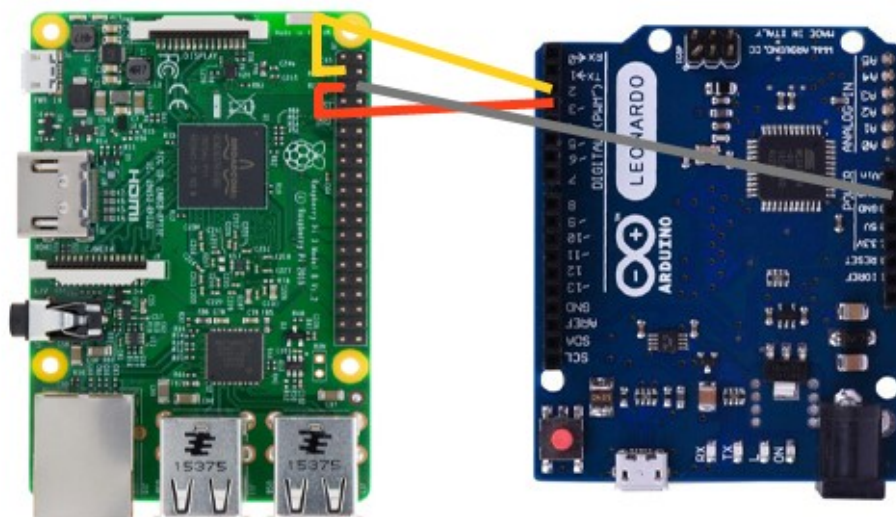
Po tomto už môže prebiehať komunikácia. Jednotlivé znaky stlačené na klávesnici z Webového rozhrania prichádzajú na server ako číslo v dátovom type String. Táto hodnota predstavuje char v decimálnej hodnote (napr.: a príde ako "97"). Následne sa táto hodnota prevádza na char pomocou nasledovného kódu:

```
char c = (char) Integer.parseInt("97");
```

Pohyb myši je odosielaný na server vo formáte "x:+51;y:-49", čo znamená, o koľko a kam sa má vzdialená myš posunúť na X,Y ose.

8.2.3 Komunikácia medzi Raspberry Pi a Arduinoom Leonardom

Pre komunikáciu medzi Raspberry Pi a Arduinoom Leonardom sa využíva I2C zbernica. V Jave využíva I2C je umožnená vďaka knižnici Pi4J. Ešte skôr ako bude možné softvérovo komunikovať medzi zariadeniami, je nutné tieto dve hardvérové dosky prepojiť. Prepojenie je znázornené na nasledujúcom obrázku.



Obrázok 8.3: *Prepojenie Raspberry Pi a Arduino pomocou I2C zbernice*

Zdroj: Vlastné spracovanie autora

Prepojenie I2C Zbernice :

- GPIO 0 (SDA) <--> Pin 2 (SDA) – (Žltá farba)
- GPIO 1 (SCL) <--> Pin 3 (SCL) – (Červená farba)
- Zem <--> Zem – (Šedá farba)

Po prepojení dosiek môžeme pomocou Javy preposielať príkazy cez I2C zbernicu pomocou nasledovného kódu:

```
I2CBus i2c = I2CFactory.getInstance(I2CBus.BUS_1);  
I2CDevice device = i2c.getDevice(I2C_ADDR);  
device.write((byte) c);
```

V predchádzajúcom kóde **I2CBus.BUS_1** označuje, ktoré I2C zariadenie využívame. Raspberry Pi obsahuje dve I2C zariadenia: /dev/ic2-0 a /dev/ic2-1. Čiže /dev/ic2-1 je BUS_1. Nasledujúcim príkazom, môžeme zistiť ktoré I2C zariadenia sú dostupné:

```
ls /dev/*i2c*
```

Ďalej v kóde vidíme premennú **I2C_ADDR** označujúcu I2C adresu pre Arduino. Je možné vybrať ľubovoľný bajt, ktorý chceme používať ako adresu Arduina (napr.: 0x8). Tento bajt sa nastavuje v kóde pre Arduino. Aby sme sa uistili, akú má Arduino I2C adresu, môžeme použiť nasledovný príkaz:

```
i2cdetect -y 1
```

Číslo 1 je označenie používaného I2C zariadenia.

```

root@student-desktop:/home/student# i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:      -- -- -- -- -- -- 08 -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
  
```

Obrázok 8.4: Príklad výsledku príkazu `i2cdetect -y 1`

Zdroj: Vlastné spracovanie autora.

Poslednou časťou kódu je **(byte) c**, kde **c** predstavuje char, ktorý získame z webového rozhrania. Podobne sa posielajú aj informácie o pohybe myši. Celý program pre Raspberry Pi je súčasťou prílohy.

8.3 Automatizácia Raspberry Pi

Automatizácia Raspberry Pi je nutná, pretože sa predpokladá, že naše zariadenie pre vzdialenú správu nebude pripájať k počítaču technik, ale neskúsený užívateľ. Preto je nutné aby sa Raspberry Pi pri svojom štarte dokázalo pripojiť na internet, na OpenVPN server a spustiť aplikáciu slúžiacu na komunikáciu.

8.3.1 Automatické pripojenie ku Internetu

Pri pripojení pomocou Ethernetového konektoru, vo väčšine prípadov netreba nič viac konfigurovať. Po detegovaní pripojenia káblu do RJ45 konektoru sa systém automaticky pokúsi o pripojenie. Pri využívaní USB LTE modemu to funguje podobne. Po prvotnom nakonfigurovaní systém automaticky deteguje, či je LTE modem pripojený, následne sa pokúsi pripojiť k internetu. Ak by bolo prístupné internetové pripojenie len pomocou Wi-Fi, môže nastať problém, pretože Raspberry Pi sa automaticky pripája len na známe Wi-Fi siete.

Pre tento prípad bolo vymyslené riešenie, kedy si Raspberry Pi SSID a heslo pre Wi-Fi získa z pripojeného USB disku. Na USB disku sa tieto informácie nachádzajú v súbore "cWifi.txt" vo formáte:

```
SSID;heslo
```


SSID, ktoré obsahujú v názve medzeru sa zapisujú ako "_". Vytvorený Bash Skript deteguje USB disk, vyhľadá "cWifi.txt" súbor, načíta SSID a heslo, pripojí sa na WiFi a následne reštartuje OpenVPN.

Vytvorený Bash Skript sa vykonáva pri štarte systému, len v prípade, že deteguje pripojený USB disk, ktorý obsahuje súbor "cWifi.txt". Aby sa skript vykonal pri štarte je potrebné ho nakopírovať do "/etc/init.d/". Celý Bash Skript je súčasťou prílohy.

8.3.2 Automatické spustenie Aplikácie pre komunikáciu

Aby pri štarte Raspberry Pi bola spustená Aplikácia slúžiaca na komunikáciu bolo potrebné vytvoriť jednoduchý Bash Skript:

```
#!/bin/bash  
java -jar /home/student/KeybMouse.jar
```

Taktiež je nutné tento skript nakopírovať do "/etc/init.d/". Potom už bude Aplikácia spúšťaná automaticky pri štarte Raspberry Pi.

8.3.3 Automatické pripojenie na OpenVPN server

Prvým krokom je nainštalovať OpenVPN na Raspberry Pi.

```
sudo apt install openvpn
```

Následne nakopírujeme konfiguračný súbor pre klienta, ktorý sme vytvorili v 4. kapitole do "/etc/openvpn/client.conf". Posledným krokom pre automatické pripojenie na OpenVPN po štarte Raspberry Pi je použitie nasledovného príkazu:

```
sudo /etc/init.d/openvpn start
```

9 Implementácia programu na emuláciu klávesnice a myši cez Arduino Leonardo

Nasledujúca kapitola sa zaoberá implementáciou kódu pre Arduino, ktoré zabezpečuje emuláciu klávesnice a myši. Emulácia klávesnice a myši je možná vďaka Arduino a dvoch knižníc, ktoré Arduino ponúka. Jedná sa o knižnicu Mouse a Keyboard. Program je napísaný v jazyku C. Pomocou I2C zbernice prijíma dáta z Raspberry Pi.

9.1 Vývojové prostredie Arduino IDE

Je multiplatformová aplikácia pre Windows, MacOS a Linux. Ide o vývojové prostredie s otvoreným kódom. Programovací jazyk je Wiring, čo je vlastne C/C++ rozšírený o knižnice. Arduino IDE je napísané v jazyku Java. Užívateľom vytvorený kód sa nazýva tzv. Sketch. Väčšinou obsahuje slučku, ktorá sa v intervaloch opakuje. Arduino IDE obsahuje dve hlavné funkcie, overenie a nahratie kódu. Viac v [29].



Obrázok 9.1: Ukážka Arduino IDE

Zdroj: Vlastné spracovanie autora

9.2 Použité knižnice

Pre emuláciu boli použité dve knižnice Keyboard a Mouse, ktoré sú oficiálne knižnice Arduina.

9.2.1 Keyboard

Táto knižnica umožňuje doskám s mikroprocesorom 32u4 posielat' klávesové skratky na pripojený počítač pomocou natívneho microUSB portu. Nie všetky znaky ASCII sú možné posielat' pomocou knižnice Keyboard. Podporované sú napríklad aj Shift alebo Caps Lock. Všetky podporované klávesy môžeme vidieť v nasledujúcej tabuľke.

Tabuľka 9.1: Zoznam podporovaných špeciálnych kláves

Klávesa	Hexadecimálna hodnota	Decimálna hodnota
KEY_LEFT_CTRL	0x80	128
KEY_LEFT_SHIFT	0x81	129
KEY_LEFT_ALT	0x82	130
KEY_LEFT_GUI	0x83	131
KEY_RIGHT_CTRL	0x84	132
KEY_RIGHT_SHIFT	0x85	133
KEY_RIGHT_ALT	0x86	134
KEY_RIGHT_GUI	0x87	135
KEY_UP_ARROW	0xDA	218
KEY_DOWN_ARROW	0xD9	217
KEY_LEFT_ARROW	0xD8	216
KEY_RIGHT_ARROW	0xD7	215
KEY_BACKSPACE	0xB2	178
KEY_TAB	0xB3	179
KEY_RETURN	0xB0	176
KEY_ESC	0xB1	177
KEY_INSERT	0xD1	209
KEY_DELETE	0xD4	212
KEY_PAGE_UP	0xD3	211
KEY_PAGE_DOWN	0xD6	214
KEY_HOME	0xD2	210
KEY_END	0xD5	213
KEY_CAPS_LOCK	0xC1	193
KEY_F1	0xC2	194
KEY_F2	0xC3	195
KEY_F3	0xC4	196
KEY_F4	0xC5	197
KEY_F5	0xC6	198
KEY_F6	0xC7	199
KEY_F7	0xC8	200
KEY_F8	0xC9	201
KEY_F9	0xCA	202
KEY_F10	0xCB	203
KEY_F11	0xCC	204
KEY_F12	0xCD	205

Zdroj: Vlastné spracovanie autora

9.2.2 Mouse

Táto knižnica umožňuje doskám s mikroprocesorom 32u4 ovládať pohyb kurzora myši na pripojenom počítači pomocou natívneho microUSB portu. Aktualizácia pozície kurzora je vždy od predchádzajúceho umiestnenia kurzora.

9.3 Program pre emuláciu klávesnice a myši

Komunikácia s Raspberry Pi prebieha pomocou I2C zbernice. Prvým krokom je nastavenie I2C adresy pre Arduino nasledujúcim kódom:

```
Wire.begin(0x8);
```

Ďalším krokom je registrovať event, ktorý reaguje na prijatie dát cez I2C zbernicu:

```
Wire.onReceive(receiveEvent);
```

Prijaté dáta môžeme prečítať pomocou príkazu:

```
Wire.read();
```

Taktiež treba začať emuláciu, a to urobíme pomocou nasledujúcich príkazov:

```
Keyboard.begin();
```

```
Mouse.begin();
```

Tieto dva príkazy musia byť zavolané pred samostatným pokusom o ovládanie pripojeného počítača. Poslednou súčasťou programu je odosielanie klávesových skratiek a pohyb myši.

- `Mouse.click(MOUSE_RIGHT)` - slúži na emuláciu pravého tlačidla myši.
- `Mouse.click(MOUSE_LEFT)` - slúži na emuláciu ľavého tlačidla myši.
- `Mouse.move(X, Y)` - slúži na pohyb kurzoru myši. X je hodnota pohybu po X-osi, Y je hodnota pohybu po Y-osi.
- `Keyboard.print(c)` - slúži na písanie znaku klávesnice.

Celý program pre Arduino je súčasťou prílohy.

10 Analýza možnosti konfigurácie sieťových zariadení

Nasledujúca kapitola sa venuje možnostiam využitia Raspberry Pi pre vzdialenú konfiguráciu sieťových prvkov.

10.1 Štandardné sieťové prvky

Jednotlivé sieťové prvky sa delia na aktívne a pasívne. Do pasívnych sieťových prvkov sa zaradzuje kabeláž. V aktívnych prvkov sú časti počítačovej siete, ktoré určitým spôsobom pracujú so signálom v sieti (zosilňujú, modifikujú, vyhodnocujú atď.). Aktívne prvky sú spravidla konkrétne zariadenia umiestené v uzloch siete.

Medzi aktívne prvky sa zaradzujú:

- Repeater (Opakovač)
- Hub (Rozbočovač)
- Bridge (Most)
- Switch (Prepínač)
- Router (Smerovač)
- Gateway (Brána)

10.2 Sieťové nástroje

Pre operačný systém Linux existuje veľmi veľa sieťových nástrojov, ktoré pomáhajú technikom sieť monitorovať, vizualizovať a nachádzať problémy.

Niektoré najpoužívanejšie nástroje:

- iftop - slúži na monitorovanie využitia siete a DNS operácií
- vnstat - umožňuje riadiť odoslané a prijaté sieťové pakety
- iptraf - zhromažďuje IP prevádzku
- nmap - sieťový skener
- tcpdump - paketový analyzátor, používa sa na analýzu komunikácie v sieti

10.3 Možnosti konfigurácie

Tým, že Raspberry Pi bude fyzicky pripojené do vzdialenej siete, môže pristupovať a konfigurovať jednotlivé aktívne prvky danej siete. Z webového rozhrania môžeme získať IP adresu Raspberry Pi a pomocou SSH²⁶ klienta PuTTY sa pripojiť do príkazového riadku Raspberry Pi a vykonať konfiguráciu. Taktiež je možné na Raspberry Pi nainštalovať DHCP²⁷ a DNS²⁸ server, ktoré sa pri konfigurácii sieťových prvkov môžu hodiť.

²⁶ Sieťový protokol určený na prihlasovanie a vykonávanie príkazov na vzdialenom počítači

²⁷ Server pre automatické prideľovanie IP adries

²⁸ Systém názvov domén

11 Dosiahnuté výsledky

Podarilo sa mi splniť všetky body zadania diplomovej práce. Pre danú problematiku som vybral správny hardvér a vytvoril som zariadenie, ktoré je schopné vykonávať vzdialenú správu, vid Obr. 11.1. Nakonfiguroval som funkčnú serverovú platformu. Vytvoril som Webové rozhranie, ktoré slúži na ovládanie výsledného riešenia. Ďalej som vytvoril dvojicu programov, ktoré slúžia na komunikáciu Raspberry Pi s Webovým rozhraním, komunikáciu Raspberry Pi s Arduinoom a následne na emuláciu klávesnice a myši. Pre možné ďalšie rozšírenie práce, bolo vytvorené Webové rozhranie ako SPA.



Obrázok 11.1: *Výsledne zariadenie*

Zdroj: Vlastné spracovanie autora

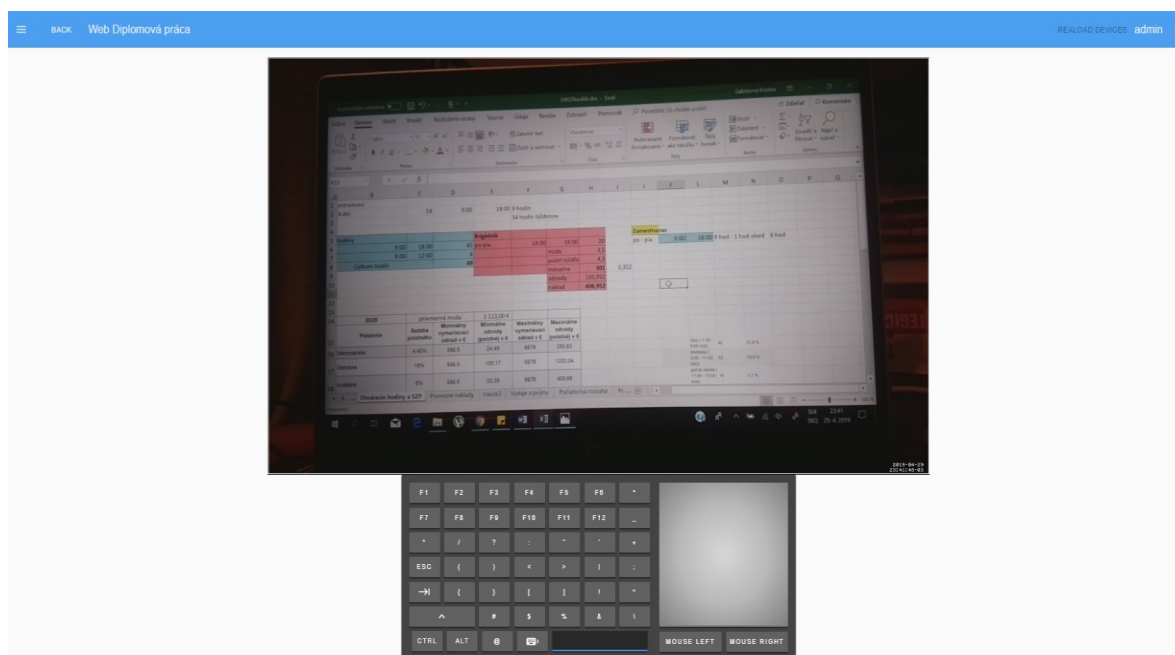
11.1 Praktická použiteľnosť

Zariadenie je plne funkčné a umožňuje vzdialene ovládať počítač. Umožňuje vzdialené sledovanie obrazu, emuláciu klávesnice a myši na pripojenom počítači a to všetko z Webového rozhrania. K internetu môže byť zariadenie pripojené pomocou Ethernetu, Wi-Fi alebo mobilnej siete. Automaticky sa pripája do OpenVPN serveru. Praktická použiteľnosť zariadenia je pomerne veľká, dokáže plnohodnotne vzdialene ovládať počítač, prípadne umožňuje konfiguráciu sieťových prvkov. Ako jeden z nedostatkov výsledného riešenia vidím, nutnosť snímania displeja pripojeného počítača pomocou kamery.



Obrázok 11.2: *Pripojenie zariadenia ku vzdialenému počítaču*

Zdroj: Vlastné spracovanie autora



Obrázok 11.3: *Ukážka z Webového rozhrania*

Zdroj: Vlastné spracovanie autora

12 Záver

Hlavným cieľom práce bolo vytvoriť zariadenie, ktoré umožní vzdialene ovládať iný počítač či zariadenie. K tomuto bolo nutné sa zoznámiť s mikropočítačom Raspberry Pi, vybrať správne periférie a zoznámiť sa s ich fungovaním, zapojením a napájaním. Ďalej bolo nutné štúdium možnosti emulovania klávesnice, popr. myši a vybrať správne zariadenie pre tento účel. Následne sa zoznámiť s fungovaním a programovaním pre mikrokontrolér Arduino.

Bolo nutné vytvoriť Serverovú platformu, na ktorej beží OpenVPN server a Webové rozhranie, ktoré umožňuje pomocou výsledného zariadenia vzdialene ovládať a vidieť iný počítač. Pre Webové rozhranie som zvolil JavaScriptový framework AngularJS a UI knižnicu s Materiál dizajnom od Googlu. Ďalej bolo nutné implementovať program slúžiaci na komunikáciu Webového rozhrania s Raspberry Pi a Arduino. Napísaná je v jazyku Java. Pre komunikáciu Raspberry Pi s Arduino pomocou I2C zbernice bola použitá knižnica Pi4J Project. Následne bolo nutné vytvoriť program pre Arduino, ktorý zabezpečuje emuláciu USB klávesnice a myši. Napísaný je v programovacom jazyku C. Vytvorené aplikácie umožňujú pomocou Webového rozhrania vzdialene sledovať displej pripojeného zariadenia a ovládať ho pomocou klávesnice a myši. Týmto splňajú cieľ práce.

12.1 Možné rozšírenia výsledného riešenia

Ako jedno z možných rozšírení je zmena snímania obrazovky cieľového počítača. Teraz sa používa na tento účel kamerový modul, čo môže byť obmedzujúce, ťažšie na manipuláciu a nastavenie. Preto by bolo vhodné použiť nejaký typ vstupnej dosky, ktorý by umožňoval Video vstup z cieľového počítača prijímať pomocou HDMI konektoru.

Ďalším možným rozšírením môže byť zautomatizovanie často opakovaných úloh. Napríklad vykonávať automatizovanú inštaláciu operačného systému, keďže jeho inštalácia je zväčša rovnaká.

Použitá literatúra

- [1] Kvmswitch. Techterms [online]. c2019 [cit. 2019-02-17]. Dostupné z: <https://techterms.com/definition/kvmswitch>
- [2] Teamviewer [online]. [cit. 2019-02-17]. Dostupné z: <https://www.teamviewer.com/>
- [3] Windows 10 how to use remote desktop. Support.microsoft.com [online]. c2019, 29. 3. 2019 [cit. 2019-02-17]. Dostupné z: <https://support.microsoft.com/sk-sk/help/4028379/windows-10-how-to-use-remote-desktop>
- [4] Chrome remote desktop. Chrome.google.com [online]. c2019, 22. 4. 2019 [cit. 2019-02-18]. Dostupné z: <https://chrome.google.com/webstore/detail/chrome-remote-desktop/gbchcmhahfdphkhkmpfmihenigjmpp?hl=sk>
- [5] Realvnc. Realvnc [online]. c2002-2019 [cit. 2019-02-20]. Dostupné z: <https://www.realvnc.com/en/>
- [6] UP Squared Specifications. Up-board.org [online]. 2018 [cit. 2019-03-05]. Dostupné z: <https://up-board.org/upsquared/specifications/>
- [7] HiKey 960. 96boards.org [online]. [cit. 2019-03-05]. Dostupné z: <https://www.96boards.org/product/hikey960/>
- [8] Odroid xu4. ODROID Wiki [online]. 17.01.2018 [cit. 2019-03-27]. Dostupné z: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>
- [9] BeagleBone Black. Beagleboard [online]. 28.06.2018 [cit. 2019-03-27]. Dostupné z: <https://beagleboard.org/black>
- [10] Tinker board. Asus [online]. [cit. 2019-03-27]. Dostupné z: <https://www.asus.com/sk/Single-Board-Computer/Tinker-Board/>
- [11] Imagination technologies ci20. Mouser [online]. c2019 [cit. 2019-03-27]. Dostupné z: <https://cz.mouser.com/new/imagination-technologies/imagination-technologies-ci20/>
- [12] JOHANSEN, Andrew. Raspberry Pi 3: The Ultimate Beginner's Guide!. CreateSpace Independent Publishing Platform, 2016. ISBN 9781540852199.
- [13] ANDERSON, Don, Dave DZATKO a Don ANDERSON. Universal Serial Bus system architecture. 2nd ed. Boston: Addison-Wesley, c2001. ISBN 02-013-0975-0.
- [14] G.KULKARNI, Udayakumar. Arduino: A Begineer's Guide [online]. 2017 [cit. 2019-04-07].
- [15] KURNIAWAN, Agus. Arduino Leonardo and Arduino Micro: A Hands-On Guide for Beginner [online]. 2016. Depok [cit. 2019-04-07].
- [16] DAYLEY, Brad. Node.js, MongoDB and AngularJS web development. Upper Saddle River, NJ: Addison-Wesley, [2014]. Developer's library. ISBN 978-013-3844-344.

- [17] GAY, Warren. Raspberry Pi hardware reference [online]. New York, NY: Apress, [2014] [cit. 2019-04-07]. Technology in action series. ISBN 14-842-0800-5.
- [18] LOMBARDI, Andrew. WebSocket: Lightweight Client-Server Communications. O'Reilly Media, 2015. ISBN 9781449369255.
- [19] Motion project [online]. [cit. 2019-04-15]. Dostupné z: <https://motion-project.github.io/>
- [20] Functions. Arduino [online]. c2019 [cit. 2019-04-15]. Dostupné z: <https://www.arduino.cc/reference/en/#functions>
- [21] Nastavení mobilních dat u českých operátorů. Lukas vlcek [online]. c2019, 25. 2. 2019 [cit. 2019-04-15]. Dostupné z: <https://www.lukas-vlcek.cz/nezarazene/nastaveni-mobilnich-dat-u-ceskych-operatoru-o2-vodafone-t-mobile/>
- [22] Debian. Debian [online]. c1997-2019 [cit. 2019-04-15]. Dostupné z: <https://www.debian.org/index.sk.html>
- [23] Openvpn. Openvpn [online]. c2019 [cit. 2019-04-27]. Dostupné z: <https://openvpn.net/>
- [24] Webstorm. JetBrains [online]. c2019 [cit. 2019-04-27]. Dostupné z: <https://www.jetbrains.com/webstorm/>
- [25] Apache. Apache [online]. 2019 [cit. 2019-04-27]. Dostupné z: <https://httpd.apache.org/>
- [26] IntelliJ IDEA. JetBrains [online]. c2000-2019 [cit. 2019-04-27]. Dostupné z: <https://www.jetbrains.com/idea/>
- [27] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. 5th ed. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
- [28] Welcome to Pi4J!. The Pi4J Project [online]. [cit. 2019-04-27]. Dostupné z: <https://www.pi4j.com/1.2/index.html>
- [29] SELECKÝ, Matúš. Arduino: uživatelská příručka. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.

Zoznam príloh

Súčasťou DP je CD/DVD.

Adresárová štruktúra priloženého CD/DVD:

- Arduino - zložka obsahuje program pre Arduino
- Motion - zložka obsahuje konfiguračný súbor pre program Motion
- OpenVPN - zložka obsahuje OpenVPN konfiguračné súbory pre Klienta a Server
- PrintModel - zložka obsahuje 3D model krabicky pre 3D tlačiareň
- RaspberryPi - zložka obsahuje program pre Raspberry Pi a skripty pre automatizáciu
- WebRozhranie - zložka obsahuje Webové rozhranie
